



CENTRO UNIVERSITÁRIO CARIOCA

CIÊNCIA DA COMPUTAÇÃO

MICHAEL CLAYTON GARCIA DA COSTA 2018200580

VINÍCIUS LIMA DA SILVA 2018200264

WENDERSON CAMPOS FERNANDES 2018102183

**OTIMIZANDO PROCESSOS: IMPLEMENTAÇÃO DO SISTEMA DE GESTÃO
NA CONCESSIONÁRIA COM CAMUNDA**

Rio de Janeiro

2023

MICHAEL CLAYTON GARCIA DA COSTA

2018200580

VINÍCIUS LIMA DA SILVA

2018200264

WENDERSON CAMPOS FERNANDES

2018102183

**OTIMIZANDO PROCESSOS: IMPLEMENTAÇÃO DO SISTEMA DE GESTÃO
NA CONCESSIONÁRIA COM CAMUNDA**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Centro Universitário Carioca, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Alberto Tavares da Silva

Rio de Janeiro

2023

MICHAEL CLAYTON GARCIA DA COSTA

2018200580

VINÍCIUS LIMA DA SILVA

2018200264

WENDERSON CAMPOS FERNANDES

2018102183

**TEMA: OTIMIZANDO PROCESSOS: IMPLEMENTAÇÃO DO SISTEMA DE
GESTÃO NA CONCESSIONÁRIA COM CAMUNDA**

Banca Examinadora

Prof. Alberto Tavares da Silva, D.Sc. - Orientador

Centro Universitário Carioca

Prof. Sergio Assunção Monteiro, D.Sc. – Professor Convidado

Centro Universitário Carioca

2023

AGRADECIMENTOS

Gostaria de expressar nossa mais profunda gratidão a todas as pessoas que tornaram possível a conclusão deste trabalho, representando não apenas uma etapa acadêmica, mas uma jornada de aprendizado e crescimento pessoal. Agradecemos de coração à nossa amada família, em especial aos nossos pais, pelo amor incondicional, apoio incansável e constante inspiração. Suas palavras de encorajamento foram a força que nos impulsionou nos momentos desafiadores. Quero expressar minha sincera gratidão a todos os professores que estiveram ao nosso lado ao longo desta jornada acadêmica. Seus ensinamentos e orientações foram fundamentais para o nosso desenvolvimento como estudante de Ciências da Computação. Não posso deixar de agradecer aos amigos e colegas que compartilharam conosco momentos de estudo, desafios e conquistas. Suas contribuições foram inestimáveis, e a camaradagem que cultivamos tornou esta jornada ainda mais significativa. Que este trabalho represente não apenas a conclusão de uma etapa, mas o início de novos horizontes e realizações.

DEDICATÓRIA

Dedicamos este trabalho aos nossos amados pais, cujo apoio incansável e amor incondicional foram cruciais, guiando cada página desta jornada acadêmica. A vocês, que são nossa fonte de inspiração e força, nosso mais sincero agradecimento.

RESUMO

A implementação de uma gestão eficaz é fundamental para qualquer organização, constituindo um elemento estratégico que almeja controlar e organizar seus produtos. Essa abordagem visa reduzir custos e minimizar erros, e a para alcançar esses objetivos, as ferramentas tecnológicas desempenham um papel crucial em uma administração. Com os avanços tecnológicos cada vez mais acessíveis, as empresas buscam novas alternativas para aprimorar a gestão de seus estoques, isso possibilita uma administração mais eficaz, promovendo maior eficiência e economia de recursos. Graças as tecnologias do século atual, houve uma notável redução nas margens de erro no controle de estoque, tem tido uma contribuição significativa no aprimoramento das atividades de contagem, reposição de produtos, organização. Consequentemente temos a diminuição das perdas, redução de custos, e a implementação de uma gestão eficiente que pode ser adotada por empresas de diferentes segmentos. Este projeto se propõe a apresentar tecnologias contemporâneas que simplificam e otimizam o trabalho de gerentes e administradores. Tendo o foco principal do projeto, aprimorar a gestão de uma concessionária, economizando tanto tempo quanto dinheiro. Além disso, o projeto visa garantir a responsividade, ou seja, a adaptação do sistema a diferentes dispositivos, como computadores, celulares, notebooks, entre outros.

Palavras-chave: Gestão de Estoque, Spring Boot, Camunda, VUE, Processos, Frameworks.

ABSTRACT

The implementation of effective management is fundamental for any organization, constituting a strategic element that allows controlling and organizing its products. This approach aims to reduce costs and minimize errors, and to achieve these goals, as technological tools play a crucial role in an administration.

With technological advances becoming more accessible, as companies seek new alternatives to improve the management of their inventories, this enables more effective administration, promoting greater efficiency and resource savings. Thanks to the technologies of the current century, there has been a notable reduction in the margins of error in stock control, there has been a significant contribution to improving concentration activities, product organization, organization. Consequently, we have the reduction of losses, the reduction of costs and the implementation of an efficient management that can be adopted by companies from different segments.

This project proposes to present contemporary technologies that simplify and optimize the work of managers and administrators. Having the main focus of the project, improve the management of a convenience, saving both time and money. In addition, the project aims to ensure responsiveness, that is, the adaptation of the system to different devices, such as computers, cell phones, notebooks, among others.

Sumário

1 INTRODUÇÃO	12
1.1 Gestão de uma concessionária	12
2.2 Objetivo do Projeto	13
1.3 Justificativa da Escolha do Tema	13
1.4 Estrutura do Trabalho.....	14
2 APRESENTANDO OS FUNDAMENTOS DO BACK-END E FRONT-END: UMA VISÃO ABRANGENTE	16
2.1 Aplicações Front-end	16
2.1.1 HTML, CSS e JavaScript: As pedras fundamentais do desenvolvimento Web	18
2.1.2 Linguagem de Marcação HTML	18
2.1.3 Linguagem de Folhas de Estilos CSS	19
2.1.4 Linguagem JavaScript	20
2.1.5 A importância do Front-end no projeto	20
2.2 Aplicações Back-end.....	21
2.2.1 Banco de dados.....	22
2.2.2 Manipulação de dados (CRUD)	23
2.2.3 Importância do Back-end no projeto	24
2.3 Funções da API no Front-end e no Back-end	25
2.3.1 Papel da API no Front-end	25
2.3.2 Papel da API no Back-end.....	25
2.4 O uso em conjunto de Front-end e Back-end.....	25
3 EXPLORANDO O MUNDO DOS FRAMEWORKS: FERRAMENTAS ESSENCIAIS PARA O DESENVOLVIMENTO DE SOFTWARE	27
3.1 Explorando fundamentos do Bootstrap.....	28
3.2 Explorando fundamentos do Vue.JS	31
3.2.1 Arquitetura MVVM.....	33
3.3 Linguagem de programação JAVA	35
3.4 Spring Framework Back-end.....	37
3.4.1 Spring Boot	40
3.5 Banco de dados H2	41
3.6 Automação de Processos de Negócios com Camunda BPM	42
4 MODELAGEM BPMN E CAMUNDA NA GESTÃO DE CONCESSIONARIA DE VEÍCULOS.....	45
4.1 Definição de requisitos	45

4.1.1 Requisitos funcionais	45
4.1.2 Requisitos não-funcionais	46
4.2 Diagrama de Atividades (BPMN)	46
4.2.1 Modelagem e notação de processos de negócios (BPMN)	47
4.3 Diagrama de casos de uso	50
4.4 Diagrama de classes	51
4.5 Descrição da Implementação	53
4.5.1 Desenvolvimento rápido com JHipster	54
4.5.2 Conexão com banco de dados	56
4.6 Interfaces com o Usuário	57
4.6.1 Protegendo senhas com JWT	59
4.6.2 Interface para BPMN	61
4.7 Métricas de aplicação.....	68
5 CONCLUSÃO.....	70
REFERÊNCIAS.....	72

LISTA DE FIGURAS

Figura 1 - Interação do usuário com Front-End e Back-End.....	16
Figura 2 - Interação do usuário GE.	17
Figura 3 – Comparação entre HTML, CSS e JavaScript.....	18
Figura 4 – HTML e CSS.....	19
Figura 5 – Fluxo Web Simples.	21
Figura 6 - Acesso aos dados de um Banco de dados.	22
Figura 7 – CRUD.....	23
Figura 8 – Back-end developer.....	24
Figura 9 – Carro back-end e front-end	26
Figura 10 - Framework desenvolvimento de softwares.....	27
Figura 11- O que é Bootstrap.....	28
Figura 12 – Componentes Bootstrap	29
Figura 13 – Documentação Bootstrap (Button).....	30
Figura 14 – Instalação do Framework Bootstrap.....	30
Figura 15 – Vue.JS Front-end Framework.	31
Figura 16 – Benefícios do Vue.JS.....	33
Figura 17 – Arquitetura MVVM.....	34
Figura 18 – Demonstração da reutilização	35
Figura 19 – Manipulação de bloco (Compiler-java-online-Jdoodle)	36
Figura 20 - Spring Eco system	37
Figura 21 – Introdução ao Spring.....	38
Figura 22 – Arquitetura Spring Boot.....	40
Figura 23 – Banco de dados H2.....	42
Figura 24 - Orquestrador Universal de Processos	43
Figura 25 – Diagrama de Atividades (BPMN)	47
Figura 26 – Start Event.....	48
Figura 27 – Service Task	48
Figura 28 – User Task.....	49
Figura 29 – Exclusive Gateway.....	49
Figura 30 – End Event.....	50
Figura 31 – Diagrama de casos de uso	51
Figura 32 – Referência de entidades.....	52
Figura 33 – Diagrama de Classes	53
Figura 34 - IntelliJ IDEA.....	54
Figura 35 – Criação com JHipster	55
Figura 36 – Instalação do Docker.....	56
Figura 37 – Conexão com banco de dados	56
Figura 38 – Conexão de Porta	57
Figura 39 – Tela inicial	58
Figura 40 – Tela de login.....	58
Figura 41 – Dados Criptografados JWT.....	59
Figura 42 – Tela do cadastro de usuário	60
Figura 43 – Exibição de usuários	60
Figura 44 – Editar usuário	61
Figura 45 – Deploy BPMN.....	62

Figura 46 – Tela Start Form.....	62
Figura 47 – Delegate de Verificação de Documento	63
Figura 48 – Tela regularizar veículo.....	64
Figura 49 – Tela Inclusão de Documento	65
Figura 50 – Tela verificar estado do veículo.	66
Figura 51 – Tela cadastrar anuncio	67
Figura 52 – Tela revisão de cadastro.	68
Figura 53 – JVM Metrics.....	69

1 INTRODUÇÃO

A Internet nem sempre foi como a conhecemos hoje, a Internet surgiu em meados dos anos 60, na época da Guerra fria que foi um período de geopolítica entre os Estados Unidos e União Soviética, com o surgimento da Internet em 1969, com o nome de *ARPANET* (*Advanced Research and Projects Agency*), o acesso a *ARPANET* era restrito a militares e pesquisadores com a intenção de criar uma rede de comunicação entre computadores, seria um dos meios de comunicação entre o departamento de defesa americano.

A partir da década de 90, foi quando a Internet se popularizou pelo mundo, atingindo novos usuários, diante disso ocorreu a criação de novas ferramentas, navegadores de Internet, redes sociais, um melhor acesso à informação. A antiga *ARPANET*, conhecida hoje por Internet, é utilizada mundialmente como uma ferramenta de trabalho, trazendo melhorias na gestão de negócios, trazendo uma organização impactante na gestão de produtos, diminuindo consideravelmente a taxa de erros, melhorando a automação na organização de estoques, otimizando consideravelmente a redução de erros e garantindo que o produto esteja em alcance quando necessário.

Segundo Kotler (1931), a Internet nos oferece novas possibilidades para as empresas conduzirem seus negócios com mais eficiência.

A partir da combinação das informações apresentadas anteriormente, surgiu a concepção de desenvolver uma aplicação que é o resultado deste projeto, um sistema de gestão de veículos. Dentro desse cenário, a aplicação em questão será desenvolvida com o propósito de auxiliar um gestor de uma concessionária na organização de sua frota de veículos.

1.1 Gestão de uma concessionária

A gestão de estoques é essencial para uma boa gestão dentro das organizações, é um ponto estratégico que visa a redução de custos e controle de produtos, incluindo também o processo de vendas e da gestão dos fluxos de informações. Em meados dos anos 1960, empresas e também organizações enxergaram o quão benéfico era a organização de seus produtos, esse método foi evoluindo ainda mais com o decorrer do tempo, com as empresas entendendo a necessidade da adaptação as novas eras, uma abordagem que visa o planejamento e controle.

Com os avanços tecnológicos e de informações atuais, é viável administrar um estoque de uma concessionária de veículos por meios das tecnologias atuais, mantendo em si a organização, simplificando a necessidade de reposição, despesas e outros aspectos, ao mesmo tempo garantindo a segurança adequada no processo. A metodologia aplicada pode ser aproveitada em outras formas de gestão, proporcionando apoio a diversas áreas relacionadas à logística.

2.2 Objetivo do Projeto

Considerando a grande importância para uma gestão de empresa, este projeto tem como seu objetivo a utilização de ferramentas e tecnologias atuais que permitem a automação de um processo, simplificando a rotina de um gestor. A aplicação em si será desenvolvida em torno de uma concessionária que visa a gestão de veículos.

À medida em que a tecnologia avança continuamente, a obtenção de suprimentos se torna cada vez mais complexas, uma gestão no mercado atual permanece em constante desenvolvimento e adaptação para satisfazer as necessidades das empresas modernas, com isso, por meios da implementação das tecnologias proposta nesse projeto, incluindo uma delas o Camunda, que desempenha um papel central na automação de processos de negócios, estaremos adotando uma tecnologia que fornece recursos essenciais para modelagem de processos em uma organização.

1.3 Justificativa da Escolha do Tema

Uma gestão eficiente tem o potencial de reduzir gastos relacionados a armazenamento, manutenção e obsolescência, isso pode ajudar a empresa a economizar dinheiro ao longo prazo, com o controle de estoque ajuda a evitar perdas e danos, protegendo os ativos da empresa. Compreender que na gestão de uma concessionária de veículos, é desempenhado o papel de administrar seus produtos, através da utilização de ferramentas é viável aprimorar e simplificar essa gestão. Essas ferramentas são cruciais para aperfeiçoar uma gestão de uma empresa, adaptando-se às atuais do mercado, com o objetivo constante de melhorar a gestão logística, incluindo a redução de erros, acidentes, necessidade de reposição de estoque, manutenção, entre outros.

Resumidamente uma gestão eficiente melhora a eficiência operacional, também tendo impacto diretamente na rentabilidade, sendo a empresa tendo a capacidade de se adaptar às mudanças no mercado.

1.4 Estrutura do Trabalho

Esse projeto tem como finalidade apresentar a criação de uma aplicação utilizando tecnologias atuais, por meio deste projeto, buscamos descrever o uso dessas ferramentas, mostrando o passo a passo desde da sua criação. Utilizando como as principais linguagens de programação Java e JavaScript.

No capítulo 2, abordaremos duas áreas distintas, porém que são complementares no nosso projeto, sendo elas o Back-End e o Front-End, nesta seção, discutiremos os conceitos fundamentais e as funções associadas a ambas, aprofundando ainda mais nossa compreensão nessas duas áreas.

No capítulo 3, exploraremos as ferramentas que desempenharão um papel fundamental na criação do nosso projeto, introduziremos a plataforma Camunda, juntamente com o uso BPMN (*Business Process Model and Notation*), proporcionando uma visão detalhada de sua origem e funcionalidades.

Para o desenvolvimento do Front-End, utilizaremos o JavaScript como linguagem principal, aproveitando suas principais bibliotecas e frameworks, como Vue.js e Bootstrap. Essas ferramentas desempenharão um papel fundamental na implementação visual e estética do projeto, permitindo uma experiência de usuário final agradável e responsiva.

Quanto ao Back-End, adotaremos a linguagem Java e nos concentraremos no uso do Spring Boot, combinando com um banco de dados H2. Essa escolha nos permitirá o desenvolvimento do lado do servidor, encarregado do processamento de dados e da lógica de negócios que impulsionarão nosso projeto.

No capítulo 4, demonstraremos a aplicação das tecnologias abordadas neste trabalho, fornecendo uma visão detalhada da solução proposta para o nosso projeto. Destacaremos a criação dos modelos BPMN que colaborarão em conjunto com a plataforma Camunda, enfatizando as etapas automatizadas do processo. Isso incluirá a geração de formulários, evidenciando como nosso projeto atuará na gestão eficiente de uma concessionária de veículos, incorporando as melhores práticas de programação.

A implementação de uma gestão eficaz é fundamental para qualquer organização, constituindo um elemento estratégico que almeja controlar e organizar seus produtos. Essa abordagem visa reduzir custos e minimizar erros, e a para alcançar esses objetivos, as ferramentas tecnológicas desempenham um papel crucial em uma administração.

Com os avanços tecnológicos cada vez mais acessíveis, as empresas buscam novas alternativas para aprimorar a gestão de seus estoques, isso possibilita uma administração mais eficaz, promovendo maior eficiência e economia de recursos.

Este projeto se propõe a apresentar tecnologias contemporâneas que simplificam e otimizam o trabalho de gerentes e administradores. Tendo o foco principal do projeto, aprimorar a gestão de uma concessionária, economizando tanto tempo quanto dinheiro. Além disso, o projeto visa garantir a responsividade, ou seja, a adaptação do sistema a diferentes dispositivos, como computadores, celulares, notebooks, entre outros.

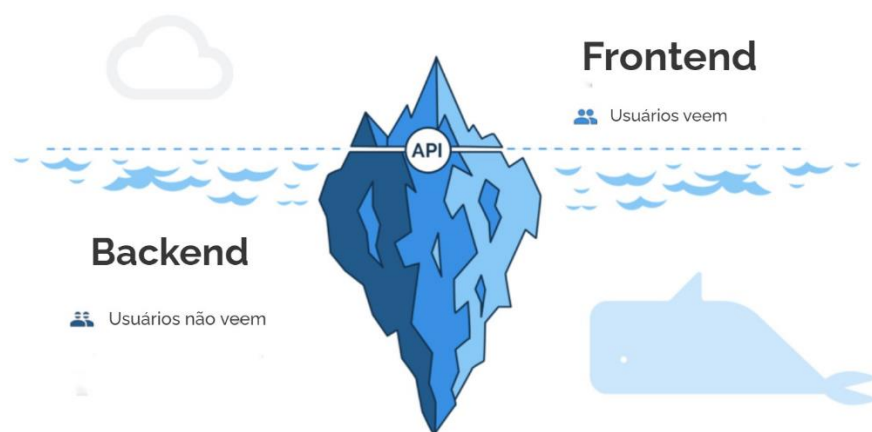
2 APRESENTANDO OS FUNDAMENTOS DO BACK-END E FRONT-END: UMA VISÃO ABRAGENTE

Neste capítulo abordaremos duas áreas distintas, porém que são complementares no nosso projeto, sendo elas o back-end e o front-end, nesta seção, discutiremos os conceitos fundamentais e as funções associadas a ambas, aprofundando ainda mais nossa compreensão nessas duas áreas.

2.1 Aplicações Front-end

Quando navegamos na Internet, interagimos com uma variedade de sites e aplicativos, e a primeira coisa que notamos é como eles são visualmente atraentes e fáceis de usar. Pode se dizer que a camada de Front-end é tudo aquilo que o usuário consegue ver e interagir na aplicação, como recursos gráficos, elementos de uma interface e até mesmo parte da lógica de programação do site. De acordo com a figura 1, temos uma simples ilustração do Front-end e Back-end.

Figura 1 - Interação do usuário com Front-End e Back-End.



Fonte: Elaborada pelo autor

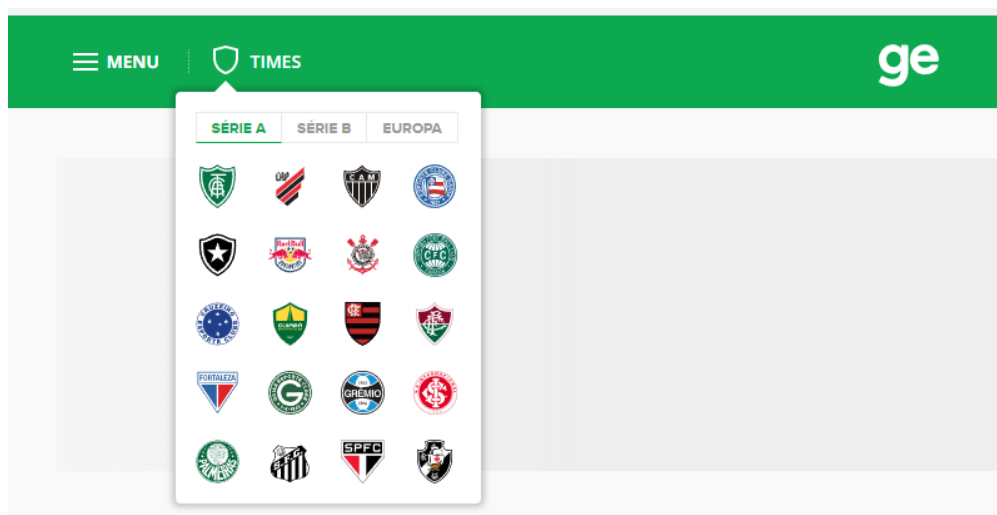
Abordando o front-end, que é a face visível e interativa de um sistema, podemos descrevê-lo de maneira mais precisa como o "lado do cliente" ou a "interface do usuário". É o componente responsável por apresentar informações e funcionalidades diretamente ao usuário final (OLIVEIRA, LUCAS, 2023). Nessa perspectiva, o usuário final interage

diretamente com o sistema, sendo a parcela visível que engloba elementos de design, conteúdo, layout e interatividade.

Essa camada é encarregada de criar elementos visuais, como botões, formulários, gráficos, design e menus, com o objetivo de proporcionar uma experiência agradável ao usuário final, isso é o que chamamos de UI (Interface de Usuário) (RAMOS, LARISSA SIQUEIRA, 2023). Essa é uma das funções essenciais do front-end. Além disso, o front-end lida com a apresentação de dados, tornando informações legíveis ao converter dados brutos em representações visuais, como tabelas e gráficos (MEDEIROS, RODRIGO, 2017).

Outro aspecto importante é a interatividade, que se refere à maneira como o usuário interage com o sistema, seja por meio de cliques do mouse, toques na interface ou outras ações. A interatividade é fundamental para a usabilidade do sistema (AWARI, 2023). Um exemplo prático disso é quando um usuário clica em um menu de navegação, nesse caso o usuário está interagindo diretamente com o sistema para realizar essa tarefa. Conforme é mostrado na figura 2, uma interação do usuário.

Figura 2 - Interação do usuário GE.



Fonte: Elaborado pelo autor

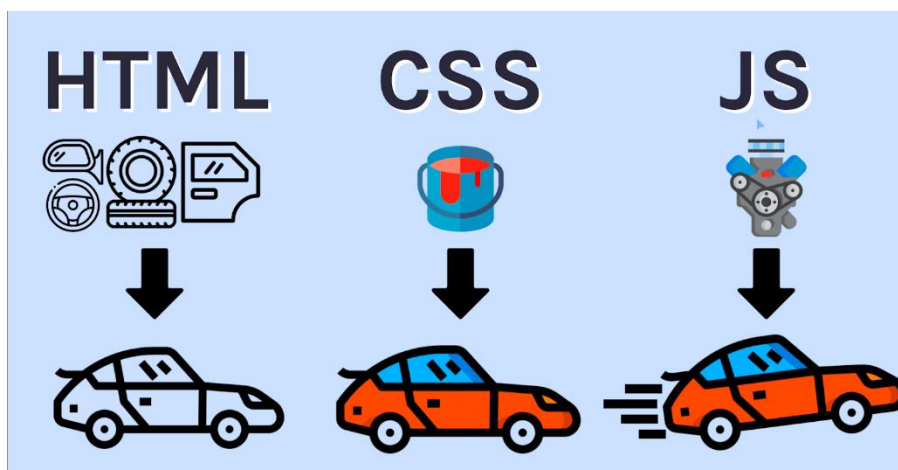
Falaremos da utilização em conjunto do back-end e do front-end, focando na perspectiva do front-end em relação à sua interação com back-end. Isso envolve uma comunicação eficiente com o servidor, buscando informações nele e apresentando esses dados de maneira compreensível para o usuário final.

2.1.1 HTML, CSS e JavaScript: As pedras fundamentais do desenvolvimento Web

Cada uma dessas tecnologias desempenha um papel fundamental no desenvolvimento web, e quando usadas em conjunto elas permitem a criação de páginas web completas e eficazes, quando usados em conjunto o HTML, CSS e JavaScript podem melhorar a usabilidade das páginas, o HTML fornecendo a estrutura semântica para o conteúdo, o CSS melhorando a aparência e tornando legível e o JavaScript adiciona funcionalidades interativas tornando a experiência do usuário mais agradável (JAYE, 2022).

Cada um desempenha um papel crucial na criação de páginas web atraentes, funcionais e interativas, como demonstrado na Figura 3 que ilustra a colaboração das três ferramentas em conjunto.

Figura 3 – Comparação entre HTML, CSS e JavaScript.



Fonte: (SCRIMBA, 2023)

2.1.2 Linguagem de Marcação HTML

O HTML (*HyperText Markup Language*) não é classificado como uma linguagem de programação, mas sim como uma linguagem de marcação. Ela é usada para estruturar o conteúdo de uma página, é nela que definimos a hierarquia dos elementos de uma página, seja ela, um título, formulário, imagens, links, entre outros.

O HTML é responsável por organizar esses elementos, sendo conhecidos como "tags" que desempenham um papel fundamental na marcação do conteúdo da página. Cada tag é composta geralmente por um nome envolto em colchetes.

HTML sendo a espinha dorsal do front-end, fornecendo a estrutura inicial básica de uma página, a base de tudo, elementos HTML podem ser aninhados dentro de outros elementos,

criando hierarquias. Resumidamente é a base da criação de páginas, permitindo uma estruturação de conteúdo, sendo o primeiro pilar.

Temos a sua versão mais recente sendo HTML5, é uma revisão significativa da Linguagem de marcação HTML sendo publicada pelo W3C (*World Wide Web Consortium*) em outubro de 2014, desde então o HTML 5 veio a se tornar a versão mais recente, sendo a versão padrão e amplamente adotada da linguagem HTML para o desenvolvimento (CARVALHO, LUANA, 2014).

Enquanto o HTML define a estrutura, o CSS assume a responsabilidade pelo design de um documento HTML, abrangendo aspectos como fontes, tamanhos, cores e muito mais.

2.1.3 Linguagem de Folhas de Estilos CSS

CSS (*Cascading Style Sheets*) É uma linguagem de estilo, responsável por controlar a aparência de uma página HTML, permitindo o uso de cores, fontes, espaçamentos entre elementos, o layout, entre outros. Ela desempenha um papel fundamental na apresentação visual de uma página WEB, temos dentro do CSS uma das suas características principais é a separação de conteúdo do HTML e do CSS, onde o HTML criamos a estrutura e no CSS criamos sua aparência, essa separação facilita e muito uma manutenção e principalmente o reaproveitamento de estilos em toda página WEB. Conforme ilustrada na figura 4.

Figura 4 – HTML e CSS



Fonte: Elaborada pelo autor

No CSS, encontramos códigos que desempenham um papel crucial na modificação do nosso HTML, influenciando a aparência por meio de alterações como tamanho, cores e a inclusão de bordas. O CSS oferece uma variedade de códigos para personalizar a aparência de elementos específicos.

Dentro dessa prática de estilização, contamos com frameworks que simplificam esse processo, como o Bootstrap e o Bulma. São ferramentas que tornam mais fácil o uso dessas estilizações. O CSS, de fato, desempenha um papel fundamental no desenvolvimento de páginas atraentes, responsivas e interativas (LOPES, 2023).

A versão mais recente do CSS é o CSS3, ele é composto por diversas especificações que foram implementadas ao longo do tempo, cada módulo do CSS3 é acompanhado por seu próprio cronograma de desenvolvimento e implementação pelos fabricantes de navegadores. Trazendo várias melhorias que contribuem para uma melhor da compatibilidade com novos navegadores.

2.1.4 Linguagem JavaScript

JavaScript é uma linguagem de programação que ajuda a tornar sites mais interativos e dinâmicos, mas também é uma linguagem aplicada em muitos outros contextos. Ela permite que os desenvolvedores adicionem funcionalidades, como responder a cliques e preenchimentos de formulários, melhorando a experiência do usuário na web. Permitindo a criação de funcionalidades na manipulação do DOM (*Document Object Model*) e as chamadas das APIs, sendo executado diretamente no navegador do usuário, o que o torna uma parte fundamental do desenvolvimento do front-end (MOZILLA, 2023). O script pode ser incorporado diretamente no HTML, mas é aconselhável utilizá-lo a partir de um arquivo externo ao HTML.

Temos o JavaScript sendo amplamente utilizado no desenvolvimento de aplicações web modernas e tem uma comunidade ativa de desenvolvedores que continuam a inovar e expandir seu alcance para além do navegador. É possível o uso do JavaScript na parte que envolve o back-end, mas optamos pelo uso da linguagem de programação Java.

No próximo capítulo (Capítulo 3), discutiremos e apresentaremos exemplos de como utilizamos frameworks para desenvolver interfaces em nosso projeto. Entre esses frameworks, destacam-se o Bootstrap e o Vue.js, que desempenharam um papel significativo na configuração do design. A linguagem principal que utilizamos é o JavaScript, que é responsável pela criação de scripts que possibilitam a interação do usuário com o sistema.

2.1.5 A importância do Front-end no projeto

Falando da importância do front-end no projeto de gestão de uma concessionária, teríamos uma experiência agradável de um gerente ao interagir com a concessionária,

incluindo a facilidade de navegação no sistema da concessionária, a pesquisa por veículos, visualizações de dados, informações dos veículos, uma experiência agradável, facilitando o monitoramento e o gerenciamento dos níveis de estoque, que é de extrema importância para um gestor, relatórios gerados pelo front-end mostrando informações e dados importantes sobre os veículos.

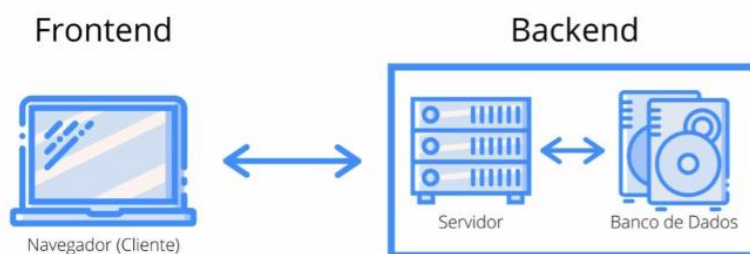
2.2 Aplicações Back-end

O back-end é a parte invisível e fundamental de um software ou aplicativo, responsável por gerenciar, processar e armazenar esses dados por trás de uma interface. É a engrenagem que permite que um sistema funcione e forneça funcionalidades para os usuários finais, incluindo o processamento de dados, armazenamento, gestão de dados que vai ser importante para nosso projeto, lógica de funcionamento (OLIVEIRA, LUCAS, 2023).

Os desenvolvedores back-end concentram-se no lado do servidor dos sites. Eles criam a estrutura e as funcionalidades geral de um site, permitindo a existência do front-end do site.

Enquanto o front-end lida com a parte visível e interativa de uma aplicação, a camada de back-end é muitas vezes chamada de “cérebro” da operação. Desenvolvedores back-end usam as APIs (*Application Programming Interfaces*) que permitem que o front-end se comunique com Back-end. A figura 5 a seguir ilustra de forma mais detalhada ao usuário como seria essa divisão.

Figura 5 – Fluxo Web Simples.



Fonte: (SOUZA, WESLEY, 2021).

A sigla API traduzindo para o português é “Interface de Programação de Aplicativos”, trata-se de um conjunto de regras e protocolos que permite que diferentes

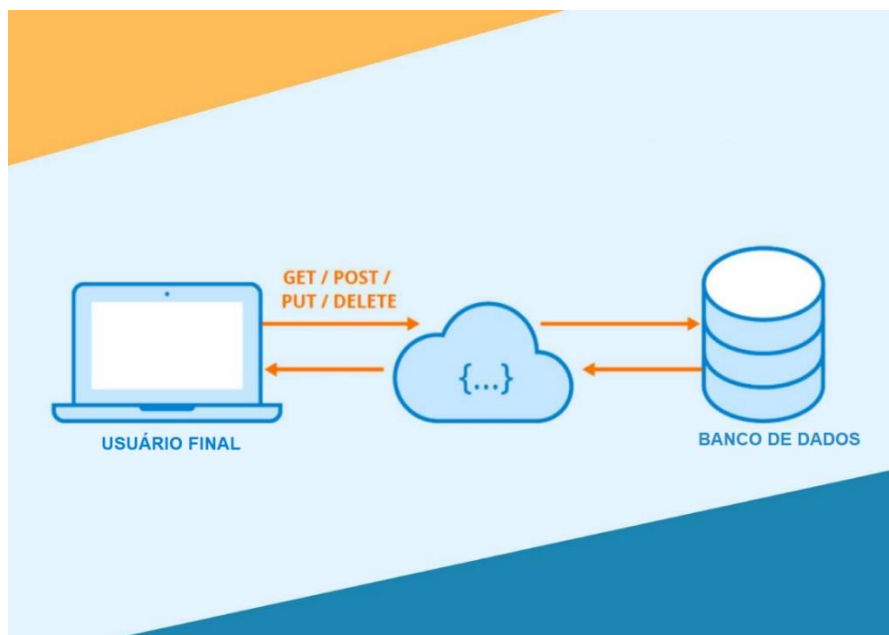
softwares se comuniquem uns com outros. Porém as APIs desempenham papéis distintos tanto no front-end quanto no back-end, no tópico 2.3 estarei exemplificando-as.

2.2.1 Banco de dados

Quando abordamos o aspecto do back-end, o banco de dados desempenha um papel vital no armazenamento, recuperação, alteração e exclusão de dados, entre outras operações. Esse processo envolve a execução de consultas SQL (Linguagem Estruturada de Consultas), a modelagem de dados e a garantia da integridade dos dados. O banco de dados é encarregado de atender às solicitações dos clientes e fornecer informações relevantes, assegurando que os dados sejam entregues aos usuários conforme necessário (KONDADO, 2022).

De maneira clara, a integração eficiente entre o banco de dados e o back-end é essencial para o funcionamento e o desempenho bem-sucedido de aplicativos e sistemas de software. Juntos, eles constituem uma base sólida para o armazenamento, recuperação e gerenciamento eficiente de dados, garantindo que os aplicativos sejam confiáveis, seguros e capazes de atender às necessidades do usuário final. Na figura 6, temos um CRUD, uma coleta de dados de um banco ao usuário final.

Figura 6 - Acesso aos dados de um Banco de dados.



Fonte: Elaborada pelo autor

2.2.2 Manipulação de dados (CRUD)

No que diz respeito à comunicação com o banco de dados, temos o CRUD, que abrevia quatro operações fundamentais para interagir com o banco de dados. Essas operações são amplamente empregadas na administração de dados, como ilustrado na Figura 7.

Figura 7 – CRUD.



Fonte: (GIAMONIANO, 2020).

Cada letra representando uma das operações:

- **Create (Criar):** essa operação envolve a criação de novas entidades no banco de dados, registros, cadastro de informações de dados de uma determinada ocasião, como por exemplo um produto que possui nome, data de validade, quantidades, entre outros. Referindo-se a criação de dados de uma tabela.
- **Read (Ler):** essa é uma operação de leitura de dados, que consiste em consultar dados existentes em um banco de dados, de modo claro seria a solicitação de dados de uma plataforma, a exibição de informações, como por exemplo, quando queremos ver a data de validade de um produto, o *Read* fica responsável por isso.
- **Update (Atualizar):** essa operação realiza a alteração e modificação desses registros criados pelo Create, permitindo modificar dados existentes no banco de dados, geralmente ela é utilizada quando precisamos atualizar alguma informação, como alterar nome do produto, alterar quantidades.
- **Delete (Excluir):** essa operação como o próprio nome já diz é responsável pela remoção desses dados do banco de dados, quando é desejado a remoção de informações que não são

mais relevantes ou necessárias, como por exemplo remover um produto de um carrinho de compras, ou eliminar o produto do banco de dados.

O CRUD é fundamental para o funcionamento de sistemas que lidam com dados, permitindo a criação, leitura, atualização e exclusão de informações de maneira organizada (SILVA, GIZELE, 2023).

2.2.3 Importância do Back-end no projeto

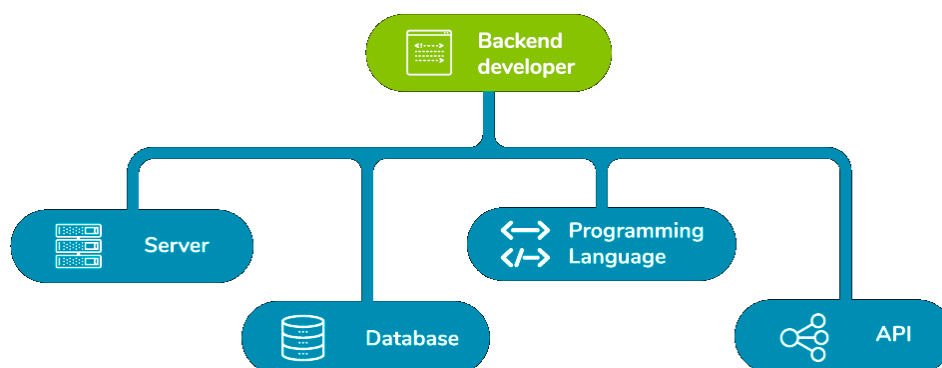
Discutindo a relevância do back-end em nosso projeto, o armazenamento de dados desempenha um papel fundamental ao gerenciar informações relacionadas aos veículos, como modelo, cor, nome, marca, ano, entre outros. A organização eficiente desse armazenamento é de suma importância para uma gestão eficaz.

Com isso teremos um controle melhor do que temos em estoque, permitindo que a gestão monitore o estoque de veículos, em resumo o back-end desempenhara um papel crucial na gestão de estoque de uma concessionária, tornando o processo mais organizado, eficiente e seguro, ele fornece uma infraestrutura necessária para gerenciar dados e automatizar processos.

No próximo capítulo iremos apresentar as ferramentas que serão utilizadas no back-end, sendo a principal dela o Spring Framework, que ficará responsável pela movimentação dos dados no nosso back-end, utilizaremos o Java como linguagem de programação.

Na figura 8, demonstraremos cada etapa do back-end.

Figura 8 – Back-end developer.



Fonte: Elaborada pelo autor

2.3 Funções da API no Front-end e no Back-end

O papel das APIs é servir como ponte de comunicação entre o front-end e o back-end, permitindo que ambas as partes trabalhem de forma eficaz e interajam de maneira coordenada, garantindo que os dados e funcionalidades sejam acessíveis de acordo com as necessidades da aplicação. Porém elas têm papéis diferentes tanto no front-end quanto no back-end de um sistema.

2.3.1 Papel da API no Front-end

Dentro do front-end, as APIs são usadas para acessar dados e serviços fornecidos pelo back-end, isso permite que o front-end exiba informações relevantes ao usuário, sendo cruciais para interação com serviços de terceiros.

Temos a manipulação de dados, as APIs permitem que o front-end manipule esses dados vindo do nosso banco de dados, e apresente ao usuário essas informações de uma forma legível e atraente, formatando e exibindo informações de forma compreendida pelo usuário.

2.3.2 Papel da API no Back-end

No back-end as APIs possuem um papel diferente, elas ficam responsáveis por expor serviços, dados ou funcionalidades para o front-end e outros sistemas externos. Temos APIs no back-end que tem a funcionalidade de ajudar a manter a segurança, aplicando autenticação e autorização para controlar quem pode acessar e executar ações específicas. Elas implementam as regras de negócios, o que pode incluir lógica sobre preços de produtos, regras de desconto, cálculos de impostos, entre outros.

É claro não podemos deixar de falar da manipulação de dados, que envolve o CRUD, onde temos APIs que ficam responsáveis por acessar, atualizar, inserir e excluir dados no banco de dados. Elas garantem a integridade dos dados e executam operações de armazenamento de recuperação.

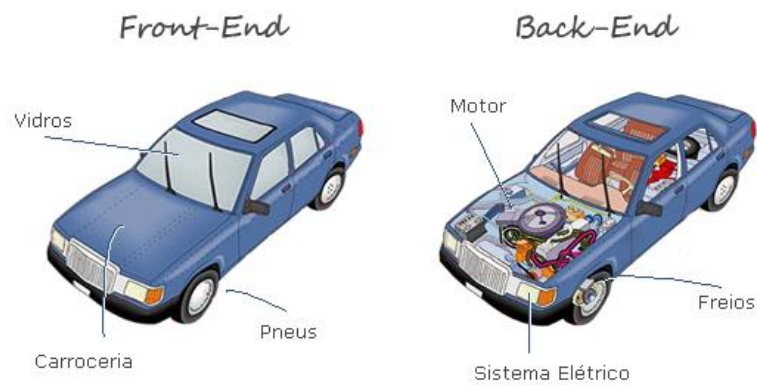
2.4 O uso em conjunto de Front-end e Back-end

O back-end e o front-end trabalham em conjunto para fornecer uma experiência de usuário final completa, de forma clara o front-end envia solicitações ao back-end para buscar dados e executar ações. O back-end processa essas solicitações e executa a lógica de negócios necessária e retorna os dados conforme foi solicitado pelo front-end, com o front-

end exibindo esses dados ao usuário de maneira atraente e interativa, permitindo que o usuário consiga interagir com o sistema.

Resumidamente o back-end cuida da inteligência por trás dos panos, enquanto o front-end fornece a interface que os usuários veem e com qual interagem, a colaboração entre essas duas partes é essencial para oferecer experiências de usuários ricas e funcionais. Conforme é ilustrado na figura 9.

Figura 9 – Carro back-end e front-end



Fonte: (SUPREMA, 2016).

3 EXPLORANDO O MUNDO DOS FRAMEWORKS: FERRAMENTAS ESSENCIAS PARA O DESENVOLVIMENTO DE SOFTWARE

Neste capítulo, discutiremos as ferramentas a serem empregadas no projeto, onde iremos resumir de forma teórica o funcionamento dessas ferramentas. No capítulo seguinte, demonstraremos sua aplicação prática e sua integração no projeto.

Um framework, é um conjunto de ferramentas, bibliotecas que ajudam os desenvolvedores a criarem aplicativos de software de forma mais rápida e eficiente. Ele fornece uma estrutura que organiza o código e oferece soluções pré-construídas para tarefas comuns, permitindo que os desenvolvedores se concentrem mais na lógica de negócios específica de seus aplicativos. Na figura 10, é ilustrada alguns desses frameworks.

Figura 10 - Framework desenvolvimento de softwares



Fonte: (CRONAPP, 2023).

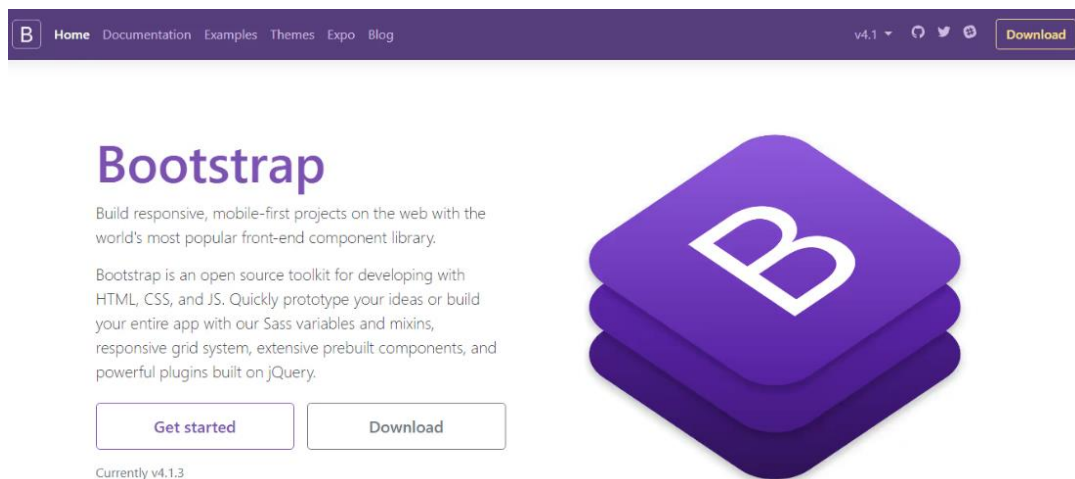
Em resumo, um framework é uma estrutura que ajuda os desenvolvedores a criarem software de maneira mais eficiente, fornecendo padrões e componentes reutilizáveis para tarefas comuns de desenvolvimento.

3.1 Explorando fundamentos do Bootstrap

O Bootstrap, também conhecido como Twitter Bootstrap devido à sua origem no Twitter, foi inicialmente desenvolvido para o Twitter por um grupo de programadores. Eventualmente, tornou-se uma das estruturas fundamentais no desenvolvimento do front-end, classificando-se como uma das mais populares em todo o mundo (GETBOOTSTRAP, 2023a).

Bootstrap é um framework de código aberto utilizado no desenvolvimento de sites e aplicações web. Ele combina CSS e Javascript para estilizar elementos de páginas escritas em HTML. É um dos frameworks mais populares do mundo. Temos uma ilustração do documento do Bootstrap, na figura 11.

Figura 11- O que é Bootstrap.



Fonte: (GETBOOTSTRAP, 2023b)

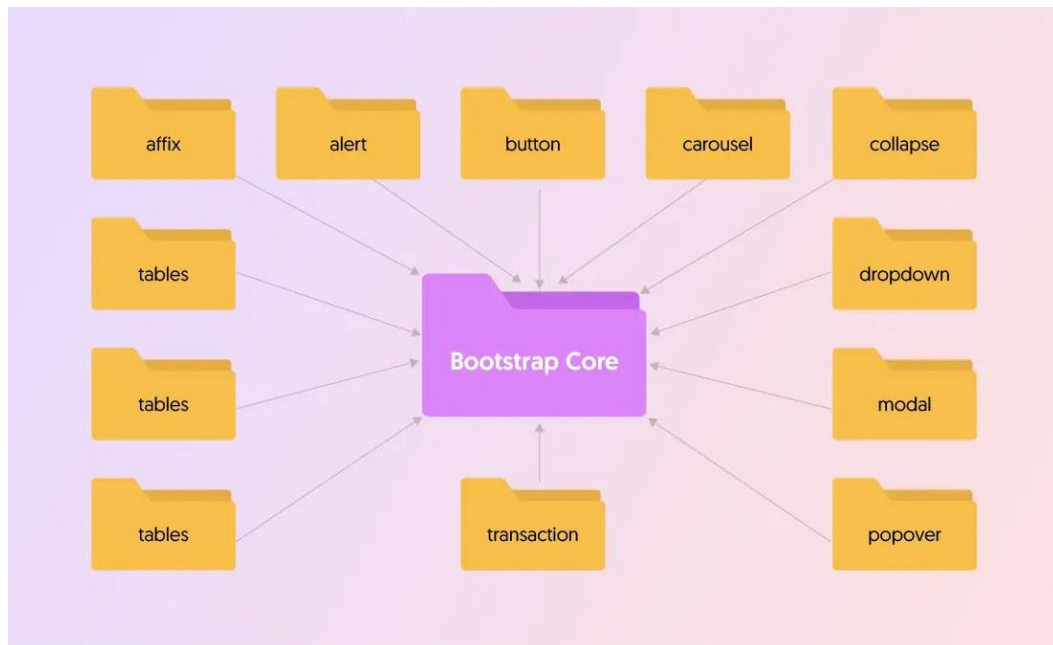
Ao trabalhar na parte visual durante o desenvolvimento de um projeto, é comum a necessidade de aplicar estilos por meio de várias classes de CSS. Nesse contexto, o Bootstrap surgiu como uma solução para estruturar e estilizar aplicações de forma eficiente. O Bootstrap é um framework de front-end que fornece um conjunto de estilos CSS para a criação simplificada e organizada de sites e aplicações.

Este framework, de código aberto, se integra perfeitamente com HTML, CSS e JavaScript, simplificando o processo de desenvolvimento de sites e aplicativos da web. Ele oferece uma extensa biblioteca de elementos HTML e componentes prontos para uso. Esses elementos foram projetados com base nas melhores práticas de design e usabilidade,

economizando tempo e esforço durante o desenvolvimento (RICARDO, HOSTINGER BRASIL, 2020).

Na figura 12 é ilustrada alguns dos seus componentes dentro do Bootstrap.

Figura 12 – Componentes Bootstrap



Fonte: (PATEL, BINAL, 2023).

O Bootstrap inclui uma ampla variedade de estilos CSS predefinidos que podem ser aplicados a elementos HTML por meio de classes. Esses estilos abrangem botões, fontes, tipografia, cores, margens, preenchimento e muitos outros. É totalmente personalizável, permitindo que o Bootstrap seja ajustado de acordo com as necessidades do projeto. Você pode selecionar componentes e estilos específicos para criar um tema personalizado que atenda às preferências do desenvolvedor.

A utilidade do Bootstrap não se restringe apenas ao CSS e HTML. Ele também oferece componentes interativos que incluem funcionalidades JavaScript, como carrosséis, modais, alertas e outros elementos interativos (GETBOOTSTRAP, 2023c). Esses componentes podem ser incorporados em seu site para proporcionar funcionalidades dinâmicas.

Destaco dois pontos principais ao usar o Bootstrap. Primeiro, o framework possui uma comunidade ativa de desenvolvedores e designers que contribuem com extensões, temas e soluções para problemas comuns. Isso torna mais fácil encontrar suporte e recursos

adicionais. Além disso, o Bootstrap é conhecido por sua documentação detalhada, que oferece uma ampla gama de exemplos práticos e informações abrangentes. Isso ajuda os desenvolvedores a começar rapidamente e explorar todas as funcionalidades da estrutura. Na figura 13, demonstramos dentro do documento do Bootstrap, um exemplo de um componente de botão.

Figura 13 – Documentação Bootstrap (Button)

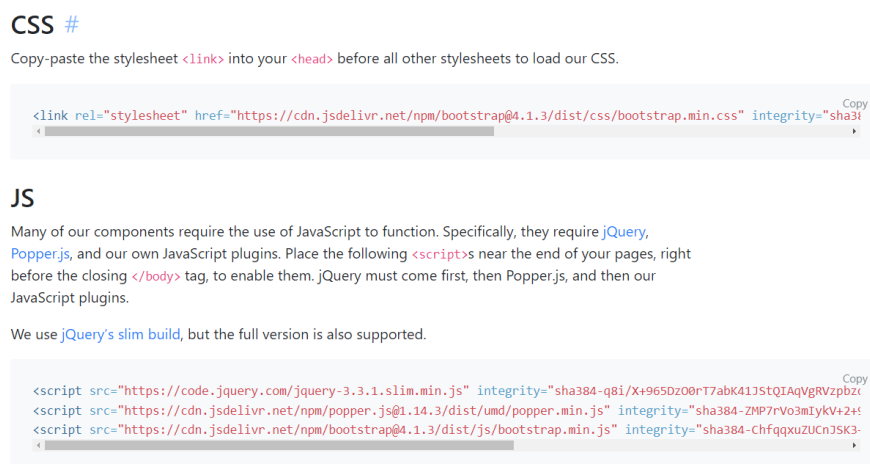


Fonte: (GETBOOTSTRAP, 2023d)

Para começar a utilizar o Bootstrap, o primeiro passo é incorporar os arquivos CSS e JavaScript do Bootstrap ao seu projeto. Em seguida, aplique as classes e componentes do Bootstrap no seu código HTML. A instalação se resume a adicionar algumas linhas de código ao seu HTML.

Onde na própria documentação detalhada nos mostra como é feito, achei interessante mostrar a sua instalação devido a facilidade, conforme é ilustrada na figura 14.

Figura 14 – Instalação do Framework Bootstrap



Fonte: (GETBOOTSTRASP, 2023e)

Em resumo, o Bootstrap permite estilizar e adicionar funcionalidades interativas a elementos HTML que já existem em seu código HTML. Ele não fornece elementos HTML prontos para uso, mas sim classes CSS e componentes JavaScript que podem ser aplicados a elementos HTML existentes. Isso possibilita a criação de páginas da web atraentes do zero sem a necessidade de escrever CSS e JavaScript.

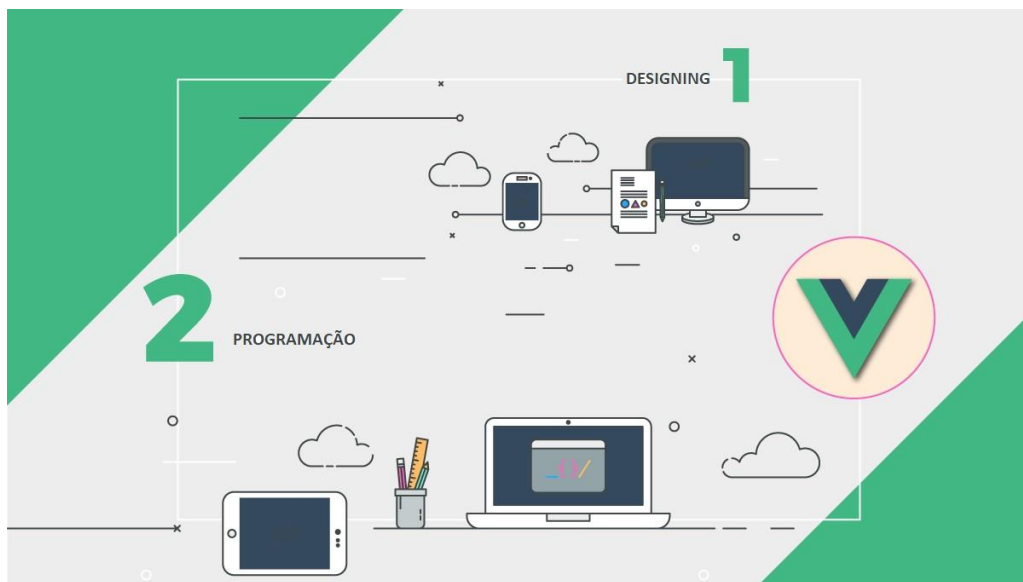
Sua popularidade se deve, em grande parte, à facilidade de uso e à capacidade de acelerar o desenvolvimento. A popularidade do Bootstrap é atribuída à sua flexibilidade, facilidade de personalização e à capacidade de criar sites responsivos de alta qualidade.

3.2 Explorando fundamentos do Vue.JS

Vue.js, também conhecido como Vue (uma abreviação), é um framework JavaScript utilizado para criar interfaces de usuário (UI) interativas e reativas. Como um framework front-end, é renomado por sua simplicidade, flexibilidade e desempenho, tornando-se uma escolha popular entre os desenvolvedores (GUEDES, 2020).

Na figura 15, temos uma pequena ilustração da separação que temos em relação a estética e a programação.

Figura 15 – Vue.JS Front-end Framework.



Fonte: Elaborada pelo autor.

Em comparação com o framework front-end Bootstrap, o Vue não consiste em um conjunto de interfaces prontas, como o Bootstrap. Em vez disso, ele oferece uma sólida

estrutura que permite criar interfaces altamente personalizadas de maneira eficaz, proporcionando as ferramentas necessárias para essa personalização.

Embora o Vue.js seja uma biblioteca JavaScript, seu foco não se restringe estritamente à linguagem JavaScript. Portanto, o Vue não é classificável como "mais voltado para o CSS" ou "mais voltado para o JavaScript". Ele integra de maneira equilibrada todas essas tecnologias. O Vue se integra estreitamente com o CSS e o HTML para criar interfaces de usuário completas e estilizadas. Ele oferece recursos para vincular estilos CSS e classes dinamicamente aos elementos da interface do usuário, permitindo um controle preciso sobre o estilo e a aparência dos componentes.

Quando se trata do Vue.js, é fundamental destacar suas principais características, que desempenham um papel crucial no processo de aprendizado. Três características de destaque dentro do Vue.js são a Reatividade, Componentes e Diretivas.

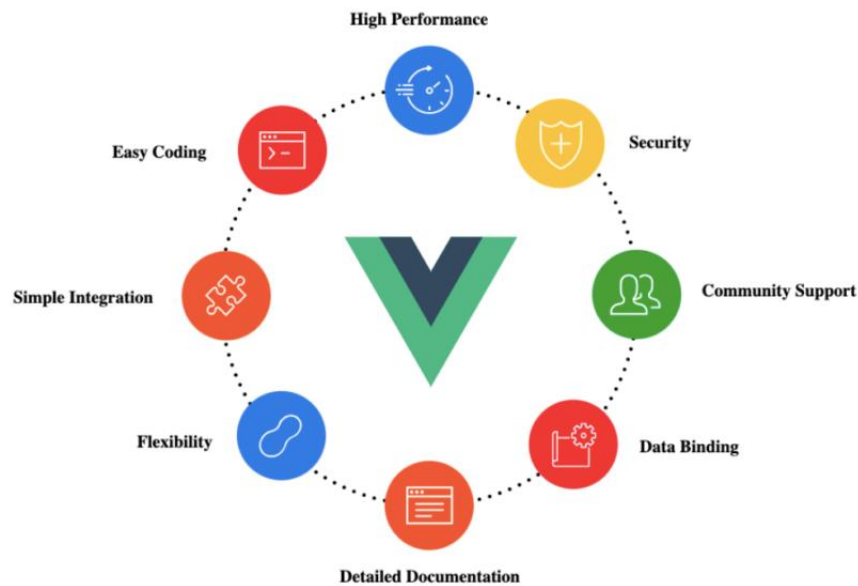
O Vue.js é renomado por seu sistema de reatividade, que permite que os dados de um aplicativo Vue sejam dinamicamente vinculados à interface do usuário. Quando os dados da nossa aplicação sofrem alterações, a interface do usuário é automaticamente atualizada, eliminando a necessidade de manipulação manual do DOM (Modelo de Objeto do Documento) (VUE.JS, 2020).

Um modelo de desenvolvimento baseado em componentes é um dos pilares do Vue.js. Nesse contexto, os desenvolvedores têm a capacidade de criar componentes reutilizáveis que englobam tanto a lógica quanto a interface do usuário. Isso simplifica a criação de interfaces complexas, combinando componentes menores de maneira eficiente.

Outro aspecto a ser considerado são as Diretivas, que expandem o HTML com atributos especiais, permitindo a vinculação dinâmica de elementos HTML a dados e a definição de comportamentos específicos (SILVA, FLAVIO OLIVEIRA, 2021).

É importante destacar que é factível criar interfaces interativas no front-end usando apenas HTML, CSS e JavaScript, porém essa abordagem não é aconselhável. A incorporação dessas bibliotecas não somente facilita o processo de desenvolvimento, mas também contribui para aprimorar a segurança, promover a reutilização e abordar diversos outros aspectos. A figura 16 ilustra os benefícios do uso da tecnologia Vue.js.

Figura 16 – Benefícios do Vue.JS



Fonte: (TWS, 2023).

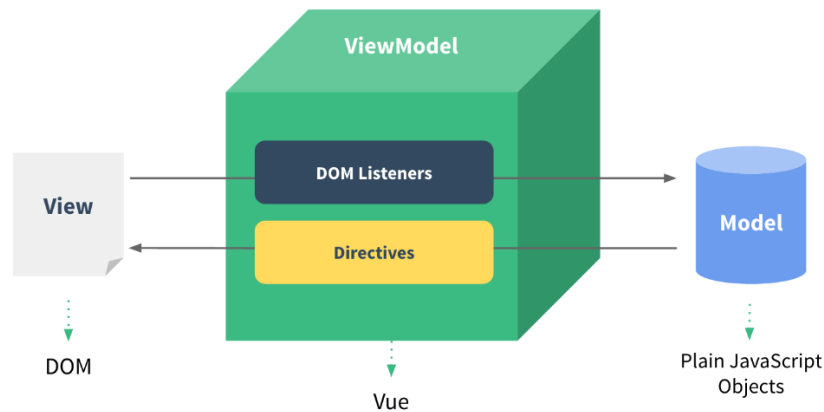
Na figura acima, gostaria de destacar três das principais vantagens do Vue.js:

- 1 **Facilidade de Aprendizado:** O Vue.js é conhecido por oferecer uma curva de aprendizado suave, tornando-o acessível mesmo para desenvolvedores iniciantes.
- 2 **Flexibilidade:** É possível criar desde pequenos componentes interativos até aplicativos de página única complexos usando o Vue.js.
- 3 **Desempenho:** O Vue.js é uma das bibliotecas front-end mais otimizadas, com foco em um desempenho eficiente, proporcionando renderização ágil e atualizações mínimas na DOM.

3.2.1 Arquitetura MVVM

A arquitetura do Vue.js é baseada no padrão MVVM (*Model-View-ViewModel*) e divide uma aplicação Vue.js em três camadas principais: Modelo (*Model*), Visão (*View*) e *ViewModel*. Cada uma dessas camadas desempenha um papel crucial na estrutura de uma aplicação Vue e contribui para a separação de responsabilidades. A figura 17, apresento como funciona essa arquitetura e irei exemplificar cada camada.

Figura 17 – Arquitetura MVVM



Fonte: (VUE.JS, 2023).

Aqui estão as características detalhadas de cada camada:

- **Model (Modelo):** A camada de modelo é responsável por gerenciar os dados e o estado da aplicação, contendo informações, dados e lógica de negócios. Esses dados no modelo podem ser originados localmente ou buscados em fontes externas, como APIs e bancos de dados. Os dados no Modelo são reativos, o que significa que qualquer alteração neles influencia automaticamente a camada de Visualização. Normalmente, esses modelos consistem em objetos ou estruturas de dados que representam o estado da aplicação.
- **View (Visão):** A camada de Visualização é encarregada de apresentar os dados ao usuário final, proporcionando acesso à aplicação. Ela é composta por elementos HTML, Templates e componentes que representam a interface do usuário. A Visualização reflete os dados do Modelo na interface do usuário, assegurando que as alterações nos dados sejam exibidas corretamente. Os Templates Vue.js são usados para definir a estrutura da Visualização, permitindo a interpolação de dados diretamente no HTML. A camada de Visualização também lida com eventos do usuário, como cliques e entradas, que são manipulados pela camada de *ViewModel*. (MARTINS, 2023).
- **ViewModel:** A camada intermediária que atua como a conexão entre as outras camadas, Modelo e Visualização. Ela representa de maneira abstrata o estado e o comportamento da interface do usuário. O *ViewModel* contém lógica de apresentação, como formatação de dados para exibição e manipulação de eventos do

usuário. Ele vincula os dados do Modelo à Visualização por meio de diretivas Vue, permitindo que os dados sejam reativos e atualizados automaticamente na interface do usuário. O *ViewModel* é uma parte essencial da reatividade do Vue.js, garantindo que os dados do modelo sejam propagados eficientemente para a Visualização (SILVA, GIZELLE CODESH, 2023).

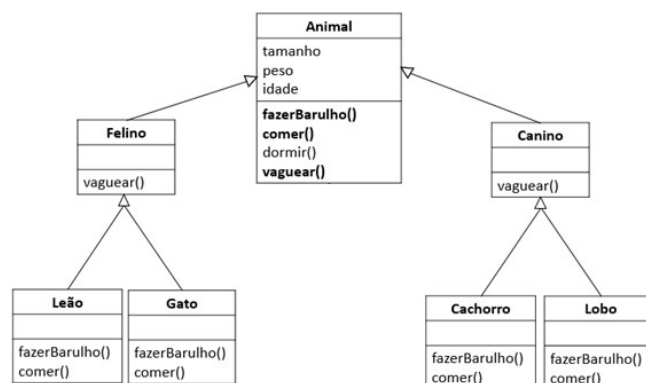
No contexto da arquitetura MVVM do Vue, essa divisão de responsabilidades ajuda a manter o código organizado, facilita a manutenção e possibilita a criação de interfaces de usuário altamente reativas.

3.3 Linguagem de programação JAVA

Antes de adentrar no framework Spring, que será empregado no projeto, é essencial discutir a linguagem de programação que serve como a base para o framework: o Java.

Java é uma linguagem de programação, à semelhança do JavaScript, que segue o paradigma de programação orientada a objetos. Desenvolvida na década de 90, ela rapidamente ganhou popularidade devido às suas características distintas e à capacidade de se adaptar a diversas áreas da computação. Java é uma linguagem de programação versátil e poderosa que se destaca pela sua natureza orientada a objetos (WIKIPÉDIA, JAVA, 2023). Isso simplifica o processo de modelagem de sistemas complexos e a criação de códigos reutilizáveis, que podem ser empregados repetidamente em todo o sistema. A seguir, apresento um exemplo bem simples de um diagrama orientado a objetos, demonstrando a modelagem e reutilização de código, onde temos métodos onde são reutilizados em outros objetos, como por exemplo os métodos “**fazerBarulho()**” e “**comer()**” conforme é ilustrada na figura 18.

Figura 18 – Demonstração da reutilização



Fonte: Elaborado pelo autor.

Uma das características essenciais do Java é sua portabilidade, o que implica que um programa Java pode ser escrito apenas uma vez e executado em qualquer sistema com uma JVM compatível (Máquina Virtual Java) (MARCELO, 2007). Além disso, a linguagem oferece segurança robusta, incluindo verificações de tipo e gerenciamento de memória, contribuindo para a prevenção de ameaças comuns, como explorações de buffer.

O Java desempenha um papel crucial em várias áreas da computação, como desenvolvimento web e desenvolvimento móvel, com o uso de tecnologias como Android. É uma escolha popular para criar aplicativos móveis em dispositivos Android. Além disso, o Java é amplamente utilizado em jogos e entretenimento. Um exemplo notável é o Minecraft, que foi originalmente desenvolvido em Java e é conhecido como “Minecraft Java Edition”. Apresento um código simples em Java que pode ser encontrado no Minecraft para manipular blocos dentro do jogo, conforme é ilustrado na figura 19.

Figura 19 – Manipulação de bloco (Compiler-Java-online-Jdoodle)

```
1 import net.minecraft.block.Block;
2 import net.minecraft.init.Blocks;
3 import net.minecraft.util.math.BlockPos;
4 import net.minecraft.world.World;
5
6 public class ExemploMinecraft {
7
8     public static void transformarBloco(World world, BlockPos pos) {
9         Block blocoAtual = world.getBlockState(pos).getBlock();
10
11         // Verifique se o bloco atual é o bloco de terra (Dirt) e substitua-o por pedra (Stone).
12         if (blocoAtual == Blocks.DIRT) {
13             world.setBlockState(pos, Blocks.STONE.getDefaultState());
14         }
15     }
16 }
```

Fonte: Elaborada pelo autor

Neste breve exemplo, o código Java realiza uma simples verificação para determinar se um bloco em uma posição específica do mundo do Minecraft é um bloco de terra (***Blocks.DIRT***). Caso seja, ele será substituído por um bloco de pedra (***Blocks.STONE***). Este foi apenas um exemplo bastante simplificado do código Java, o qual achei relevante apresentar no contexto de jogos.

O Java é aplicado em diversas áreas de desenvolvimento, mas se destaca notavelmente no âmbito do back-end, onde sua portabilidade, segurança, confiabilidade e riqueza de recursos o tornam uma escolha sólida para o desenvolvimento de sistemas de alta qualidade e alto desempenho. Há uma ampla variedade de frameworks e bibliotecas de back-end disponíveis, como Spring, Hibernate, Apache Struts, entre outras, que aceleram o desenvolvimento e simplificam tarefas comuns.

3.4 Spring Framework Back-end

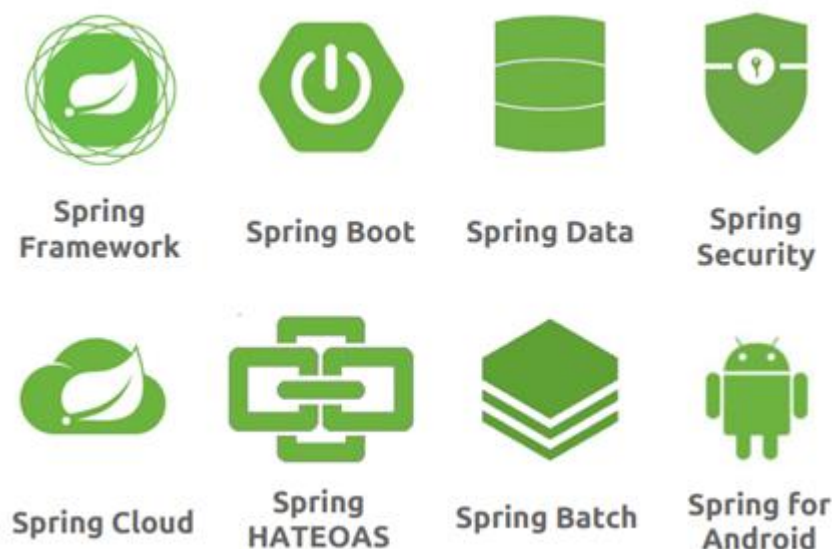
O Spring Framework é um framework de desenvolvimento de aplicações Java robusto e amplamente utilizado, com sua funcionalidade tendo o foco no back-end. Ele fornece uma ampla variedade de recursos para simplificar o desenvolvimento de aplicações empresariais.

Alguns dos recursos principais do Spring Framework que utilizaremos dentro do projeto incluem:

- **Spring Boot:** Simplifica a configuração e implantação de aplicativos Java, reduzindo a necessidade de configurações iniciais.
- **Spring Data:** Facilita o acesso aos dados da aplicação, oferecendo suporte para JDBC e JPA, entre outros.
- **Spring MVC:** Permite o desenvolvimento de aplicações usando o padrão de arquitetura MVC (*Model-View-Controller*).
- **Spring Security:** Gerencia a autenticação e autorização em aplicativos Spring.

Na figura 20, temos uma ilustração de algumas ferramentas presentes dentro do Spring Framework.

Figura 20 - Spring Eco system

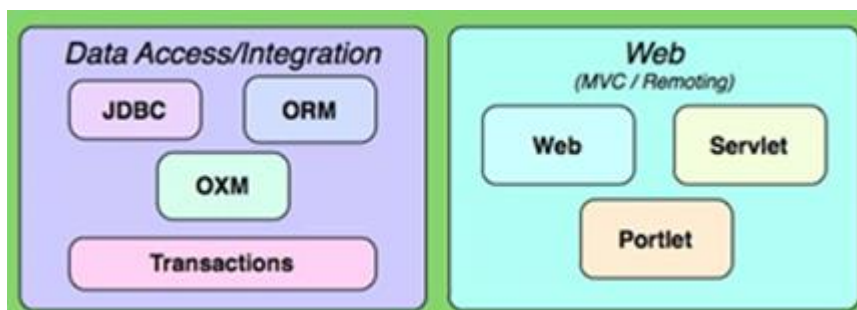


Fonte: (ECOSISTEMA SPRING, 2023)

Antes de entrarmos no Spring Boot vamos apresentar a arquitetura do Spring Framework resumindo o funcionamento de algumas camadas importantes.

Vamos abordar brevemente alguns recursos disponíveis no Spring Framework. Não abordaremos todas, uma vez que a biblioteca Spring é extensa, contendo centenas de ferramentas. No entanto iremos apresentar alguns dos recursos disponíveis no Spring, conforme é ilustrado na figura 21.

Figura 21 – Introdução ao Spring



Fonte: Elaborado pelo autor.

Começando com *Data Access/Integration*, trata-se de uma camada que engloba a obtenção de dados e o mapeamento objeto-relacional.

- O módulo JDBC (Java *Database Connectivity*) em relação à conexão e interação com bancos de dados. Este módulo nos fornece uma camada de abstração do JDBC, eliminando a necessidade de lidar com a tediosa codificação JDBC e análise de códigos de erro específicos do fornecedor do banco de dados.
- O ORM (*Object Relational Mapping*) é uma ferramenta de desenvolvimento projetada para reduzir a complexidade de programação orientada a objetos ao utilizar bancos de dados relacionais. Ele oferece integração com APIs de mapeamento objeto-relacional, com diversas ferramentas ORM amplamente utilizadas e bem conceituadas no desenvolvimento de aplicativos Java, incluindo o Hibernate, Spring Data JPA, MyBatis, TopLink, entre outras.
- O OXM (*Object XML Mapping*) refere-se à capacidade de mapear objetos Java para documentos XML e vice-versa. Isso é especialmente útil quando é necessário converter dados entre representações de objetos Java e estruturas de dados XML.
- TRANSACTIONS refere-se à administração de transações no controle de operações que englobam múltiplas etapas e precisam ser executadas como uma unidade atômica. Esse gerenciamento é efetuado através do Spring Transaction Management e pode ser implementado por meio de anotações, como a anotação `@Transactional`.

No contexto do Spring, é possível reverter uma transação, o que é conhecido como Rollback. Para ilustrar, imagine uma operação que envolve várias etapas. Após concluir a primeira etapa, avançamos para a segunda e, em seguida, para a terceira. Entretanto, se ocorrer um erro na terceira etapa, todas as ações realizadas nas etapas anteriores serão desfeitas, ocorrendo um Rollback (SPRING FRAMEWORK, 2023).

- Quanto à camada Web temos o módulo Web do Spring Framework, que nos oferece recursos fundamentais de integração orientados para a Web. Isso inclui funcionalidades para o upload de arquivos e fornece ferramentas poderosas para criar aplicativos da web seguros, escaláveis e flexíveis. O Spring Web suporta vários protocolos, estilos de arquitetura e requisitos de negócios.
- Servlets no desenvolvimento web com o Spring Framework proporciona uma abordagem estruturada e flexível para a criação de aplicativos da web em Java (DEV MEDIA, 2023). Ele incorpora a implementação do padrão MVC (*Model View Controller*) do Spring para aplicações web, onde as funcionalidades são dedicadas a lidar com solicitações HTTP e fornece recursos de desenvolvimento web. O mapeamento de URLs é feito para direcionar as solicitações aos controladores, responsáveis por processar as solicitações e executar a lógica de negócios, preparando as respostas. O Spring abstrai grande parte da complexidade associada ao tratamento de solicitações HTTP, fornecendo uma estrutura MVC para um desenvolvimento web mais organizado e eficiente.
- Portlets representa uma extensão que simplifica o desenvolvimento de Portlets em ambientes de portal web. Ele oferece recursos como o padrão MVC, injeção de dependências e suporte a eventos, auxiliando os desenvolvedores na criação de Portlets eficientes e reutilizáveis. O Spring Portlets, em particular, facilita a criação de Portlets ao fazer uso do padrão Spring MVC (*Model-View-Controller*) (FERREIRA, QUITUMBA, 2022).

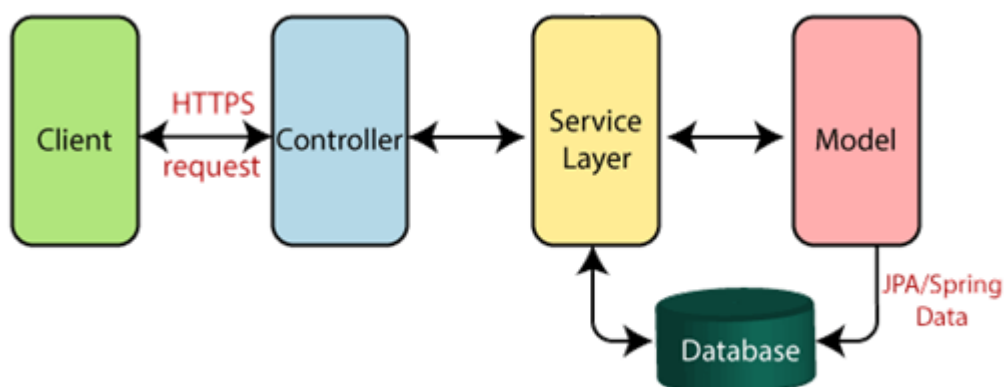
Embora o Spring framework tenha sido criado justamente com o intuito de simplificar as configurações para aplicações web, ele não atendeu 100% as expectativas do mercado ao ser lançado, já que as configurações seguiam grandes e complexas demais. Sendo assim, um novo projeto foi acrescentado ao framework para mudar esse jogo e abstrair toda a complexidade que uma configuração completa pode trazer: o Spring Boot.

3.4.1 Spring Boot

O Spring Boot é um projeto do ecossistema do Spring Framework que simplifica o desenvolvimento de aplicações em Java, ele foi criado no intuito de tornar mais fáceis a criação de aplicações mais robustas, altamente configuráveis e prontos para produção. Sendo o Spring boot projeto *open source* (código aberto) que veio para revolucionar o desenvolvimento de aplicações Java, tornando o desenvolvimento mais produtivo e permitindo que os desenvolvedores se concentrem em escrever código de negócios (IBM, 2023).

O Spring Boot nos oferece um conjunto de ferramentas que permitem aos desenvolvedores começar rapidamente, com configurações padrão ao contexto de um sistema, melhorando a produtividade de uma aplicação. Onde adotamos uma abordagem de convenção sobre configuração, sendo que por padrão o Spring Boot já nos fornece essas configurações padrões, com isso temos o foco maior na concentração na lógica de negócios em vez de perder tempo em configurações muito complexas, eliminando grande parte da complexidade associada ao desenvolvimento Java, oferecendo padrões inteligentes que reduzem também a sobrecarga de configurações e aceleram o processo do desenvolvimento, na figura 22 é mostrado como é a arquitetura do Spring Boot.

Figura 22 – Arquitetura Spring Boot



Fonte: Elaborada pelo autor.

Para resumir o funcionamento no Spring Boot por meio dessa estrutura, vejamos o que ocorre:

- **Client:** Esta camada não faz parte da estrutura do Spring Boot, mas é uma parte externa que interage com a aplicação. Normalmente, o cliente é um navegador da web ou um aplicativo móvel que envia solicitações ao servidor Spring Boot por meio de URLs.
- **Controller:** A camada de controle é responsável por receber as solicitações do cliente, encaminhar essas solicitações para a lógica apropriada e fornecer as respostas. Os controladores são componentes do Spring Boot que tratam da interação entre o cliente e a aplicação. Eles geralmente contêm métodos associados a URLs específicas e gerenciam a lógica de negócios.
- **Service:** A camada de serviço é onde a lógica de negócios da aplicação é implementada. Os serviços são componentes que encapsulam as funcionalidades de negócios e são invocados pelos controladores. Eles podem executar operações como validação, processamento e interações com o banco de dados.
- **Model:** O modelo representa a estrutura de dados da aplicação. Ele define como os dados estão organizados e representa os objetos da aplicação. Isso pode incluir classes Java que mapeiam os dados do banco de dados e objetos usados para transferir dados entre as camadas.
- **Database:** Esta camada representa o repositório de dados onde as informações da aplicação são armazenadas. Os bancos de dados podem ser de diferentes tipos, como SQL (relacional) ou NoSQL (não relacional), dependendo dos requisitos da aplicação.

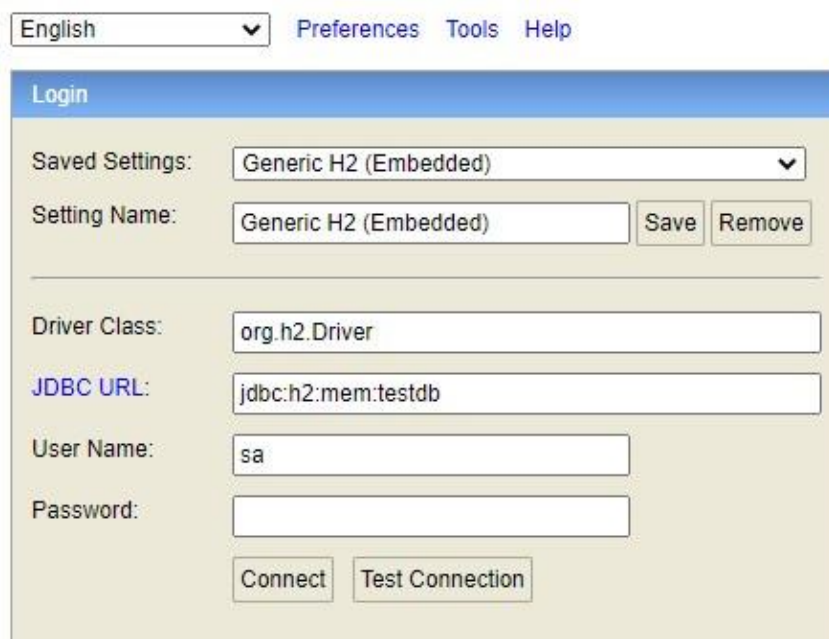
O Spring Framework representa um robusto e amplamente empregado framework para o desenvolvimento de aplicações Java. Ele disponibiliza uma diversidade de funcionalidades para facilitar a criação de aplicações corporativas (JAVAGUIDES, 2023).

3.5 Banco de dados H2

O Banco de dados H2 é uma solução de banco de dados relacional projetada para armazenar dados em tabelas com esquema definido. Ele oferece recursos comuns a bancos de dados relacionais, incluindo suporte a SQL, transações e relacionamentos entre tabelas. O H2 é uma opção excelente para o desenvolvimento e testes de aplicativos devido à sua natureza leve e à capacidade de ser incorporado em aplicativos Java.

A figura 23 exibe a tela de menu deste banco de dados, permitindo-nos acessá-lo.

Figura 23 – Banco de dados H2



The screenshot shows the H2 database login window. At the top, there is a language dropdown menu set to 'English' and three menu items: 'Preferences', 'Tools', and 'Help'. Below this is a 'Login' tab. The 'Saved Settings' section shows a dropdown menu with 'Generic H2 (Embedded)' selected. Below this, the 'Setting Name' is also 'Generic H2 (Embedded)', with 'Save' and 'Remove' buttons. The 'Driver Class' field contains 'org.h2.Driver'. The 'JDBC URL' field contains 'jdbc:h2:mem:testdb'. The 'User Name' field contains 'sa'. The 'Password' field is empty. At the bottom, there are 'Connect' and 'Test Connection' buttons.

Fonte: Elaborado pelo autor

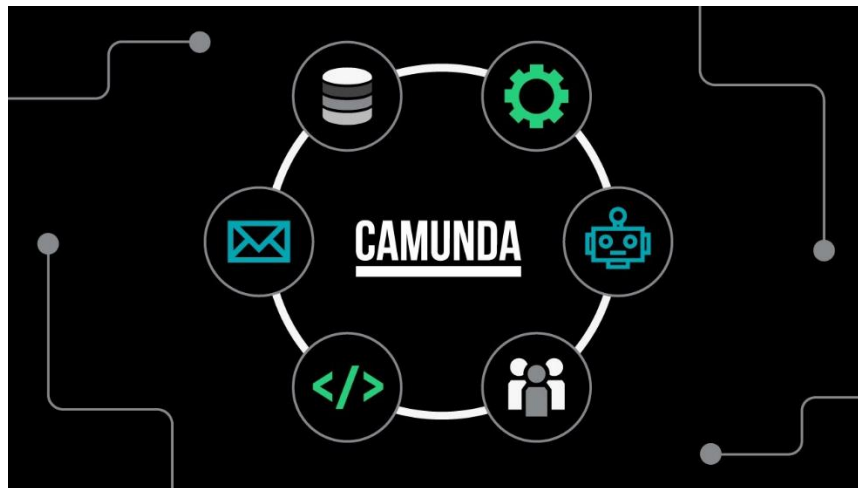
O H2 é um banco de dados desenvolvido em Java que se destaca por sua alta performance, leveza e notável flexibilidade. Além disso, ele oferece suporte à criação de bancos de dados em memória, que são temporários e existem somente durante a execução da aplicação. Essa característica é extremamente útil em contextos de testes unitários e desenvolvimento (WIKIPÉDIA H2-DATABASE, 2023).

3.6 Automação de Processos de Negócios com Camunda BPM

Camunda é uma plataforma de automação de processos de negócios, de código aberto que permite que os desenvolvedores automatizem, gerenciem e otimizem seus processos de negócios de maneira eficiente. Ela é uma das principais ferramentas de código aberto para modelagem, execução e otimização de processos de negócios.

Na Figura 24, podemos observar a presença da Camunda, uma plataforma líder na orquestração de processos de negócios, reconhecida por sua notável flexibilidade que permite que empresas de todos os setores automatizem, gerenciem e otimizem seus fluxos de trabalho de maneira eficaz.

Figura 24 - Orquestrador Universal de Processos



Fonte: (THE UNIVERSAL PROCESS, 2022).

Baseada em Java ela fornece um fluxo de trabalho inteligente ou um sistema de gerenciamento de processos de negócios (BPM) e configura uma arquitetura para codificar na linguagem de sua escolha. Aqui estão alguns aspectos importantes relacionados à Camunda:

- **Modelagem de Processos:** A Camunda fornece uma interface gráfica para a modelagem de processos de negócios usando a notação BPMN (*Business Process Model and Notation*). Isso permite que os usuários desenhem diagramas de processo que representem visualmente o fluxo de trabalho da organização.
- **Motor de Processo:** Um dos elementos essenciais da plataforma Camunda é o seu motor de processo, que assume a responsabilidade de executar e gerenciar os fluxos de trabalho e processos automatizados.
- **Integração de Sistemas:** A Camunda é altamente flexível e pode ser integrada a sistemas e aplicativos existentes por meio de APIs (*Application Programming Interfaces*) e serviços web. Isso facilita a automatização de processos que envolvem interações com outros sistemas.
- **Open Source:** A Camunda é uma plataforma de código aberto, o que significa que seu código-fonte está disponível gratuitamente para a comunidade. Isso torna a plataforma acessível a uma ampla gama de organizações e desenvolvedores.
- **Comunidade Ativa:** O Camunda possui uma comunidade ativa de desenvolvedores e usuários que contribuem para o desenvolvimento da plataforma e fornecem suporte técnico.

Camunda oferece uma variedade de benefícios, desde melhorias na eficiência operacional até a conformidade com regulamentos e a capacidade de se adaptar às mudanças do mercado. (ROSILENE ZARDO, 2023).

Amplamente utilizada em diversas indústrias e setores para melhorar a eficiência operacional, reduzir custos e aumentar a agilidade organizacional. Ela oferece uma abordagem flexível para a automação de processos, o que a torna adequada para uma variedade de cenários empresariais.

4 MODELAGEM BPMN E CAMUNDA NA GESTÃO DE CONCESSIONARIA DE VEICULOS

A utilização da modelagem BPMN em conjunto com a plataforma Camunda possui um impacto relevante na eficácia da administração de revendedoras de automóveis. Este estudo apresenta uma perspectiva ampla da solução sugerida, evidenciando a importância da automatização de procedimentos para melhorar as atividades e os atendimentos em concessionárias de veículos. Todo o conteúdo do nosso projeto está acessível no GitHub, por meio do seguinte link <<https://github.com/CostaMichael09/ProjectTcc>>.

4.1 Definição de requisitos

Os requisitos do projeto foram definidos ao início do projeto, e ao longo de reuniões, definimos no começo seus requisitos funcionais e requisitos não funcionais, e com o tempo de produção alteramos onde julgamos necessário.

4.1.1 Requisitos funcionais

- RF1: O sistema deve fornecer uma tela de login, onde os funcionários da concessionária podem se autenticar com nome de usuário e senha.
- RF2: Após o login bem-sucedido, o sistema deve permitir o cadastro de um veículo, incluindo informações como Renavam e placa.
- RF3: O sistema deve permitir que os funcionários autorizados cadastrem carros no sistema, incluindo campos para marca, modelo e ano de fabricação.
- RF4: O sistema deve ser capaz de gerar aleatoriamente uma "multa fictícia" para um carro cadastrado.
- RF6: O sistema deve permitir que um funcionário da concessionária avalie o estado físico do veículo e registre a aprovação ou reprovação do seu estado físico.
- RF7: Em caso de reprovação do seu estado físico, ele só poderá passar para o próximo estágio após resolver o problema.
- RF8: O sistema deve permitir o upload de fotos do veículo.
- RF9: As fotos devem ser revisadas e avaliadas por um funcionário da concessionária antes de permitir o avanço para o próximo estágio.
- RF10: Após a aprovação de todas as etapas anteriores, o veículo pode ser listado como disponível na concessionária.

4.1.2 Requisitos não-funcionais

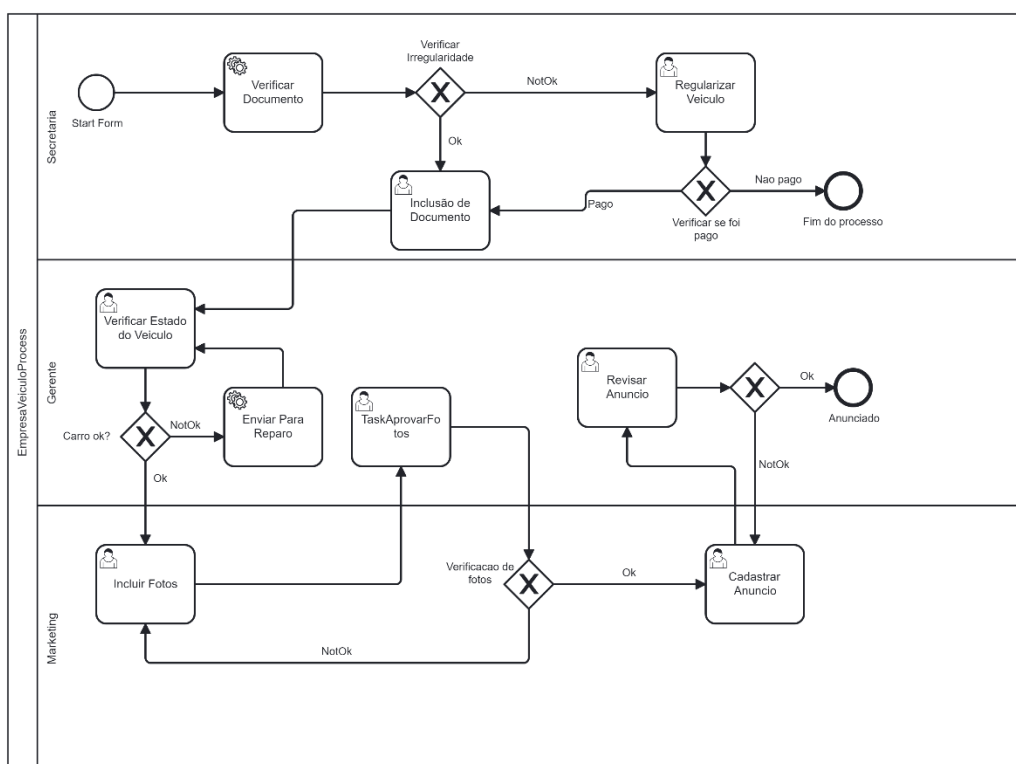
- RNF1: O sistema deve ser capaz de lidar com uma grande quantidade de registros de carros e fotos, garantindo tempos de resposta rápidos durante as operações diárias.
- RNF2: O acesso ao sistema deve ser protegido por autenticação, garantindo que apenas funcionários autorizados possam cadastrar carros e aprovar informações.
- RNF3: A interface do usuário do sistema deve ser intuitiva, permitindo que os funcionários cadastrem carros, inspecionem fotos e aprovelem informações com facilidade.
- RNF4: O sistema deve ser altamente confiável, minimizando a ocorrência de falhas ou erros durante as operações.
- RNF5: O sistema deve ser dimensionado de forma eficaz para lidar com o crescimento do volume de dados ao longo do tempo.
- RNF6: O sistema deve fornecer tempos de resposta rápidos durante todas as operações, incluindo a verificação de multa, e upload de fotos.
- RNF7: O sistema deve ser operado em qualquer navegador com suporte a HTML5 e JavaScript.
- RNF8: A linguagem do servidor utilizado no sistema será o Java.
- RNF9: O banco de dados utilizado no sistema será o H2.

4.2 Diagrama de Atividades (BPMN)

Primeiramente, vamos apresentar o diagrama de atividades, que é uma representação gráfica de um processo, também conhecido como fluxo de trabalho. Utilizamos a ferramenta denominada Camunda Modeler na versão 7.5 para este propósito. Com o auxílio dessa ferramenta, criamos o BPMN, o que nos permite exemplificar com maior clareza que o BPMN é uma notação amplamente adotada para representar processos de negócios e atividades (CAMUNDA, 2023).

O BPMN é uma ferramenta valiosa para a documentação, uma vez que esse diagrama retrata de maneira eficaz as atividades, decisões e sequências de interações em um processo, proporcionando uma compreensão nítida da dinâmica do sistema. Na figura 25, encontramos o nosso diagrama de atividades e o BPMN que foi desenvolvido no Camunda Modeler.

Figura 25 – Diagrama de Atividades (BPMN)



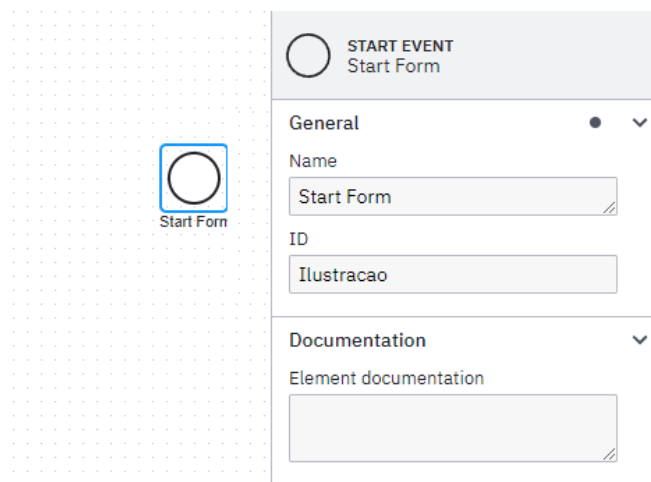
Fonte: Elaborada pelo autor

O nosso BPMN, conforme demonstrado na figura 25, consiste em um diagrama de Atividades. Vamos agora simplificar um pouco a sua utilização, oferecendo uma breve explicação das caixas que empregamos no projeto.

4.2.1 Modelagem e notação de processos de negócios (BPMN)

- *Start Form*: no âmbito do BPMN, esta expressão designa um formulário que se encontra vinculado ao começo de um processo de negócios. O conceito ao "*Start Form*" é possibilitar que os utilizadores forneçam os dados iniciais necessários para dar início a um processo de negócios. Tais informações podem englobar dados contextualmente relevantes e uma maneira de coletar dados de maneira estruturada antes do início do processo, conforme ilustrado na figura 26.

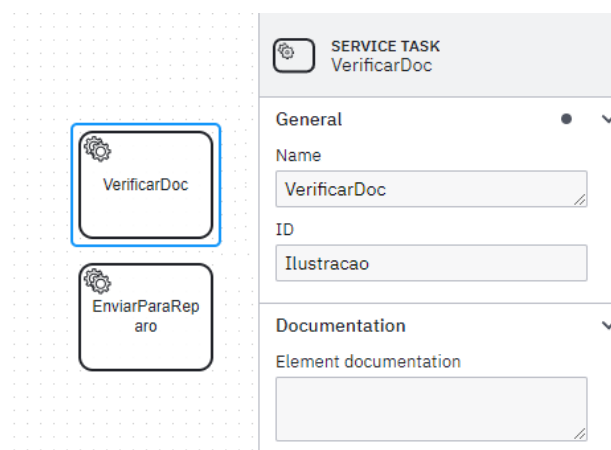
Figura 26 – Start Event



Fonte: Elaborada pelo autor.

- *Service Task*: Aqui, temos a caixa representada com uma engrenagem, como no caso de "VerificarDOC" e "EnviarParaReparo", conforme apresentado no BPMN da figura 25. Essa caixa é responsável por definir o comportamento específico que deve ser executado quando uma tarefa é acionada durante a execução do processo. Também, temos o "*delegate*", que se refere a uma classe Java, uma classe que encapsula a lógica de negócios que deve ser executada quando uma Service Task é acionada durante a execução do processo. Para esclarecer, a conexão entre o "*delegate*" e a "*Service Task*" é estabelecida ao associar o "*delegate*" à "*Service Task*", especificando o que ocorre quando a tarefa é executada, conforme ilustrado na figura 27.

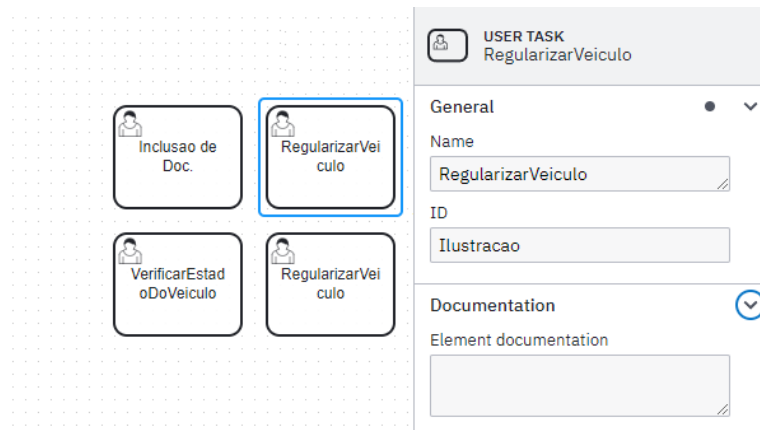
Figura 27 – Service Task



Fonte: Elaborada pelo autor.

- *User Task*: Tarefa de Usuário: é um elemento dentro do BPMN que representa uma tarefa destinada à execução por um usuário. A execução de uma “*User Task*” envolve a interação direta de um usuário, que é responsável por executar a ação definida pela tarefa, como incluir dados, preencher formulários, tomar decisões, enviar aprovações, entre outras ações, conforme ilustrado na figura 28.

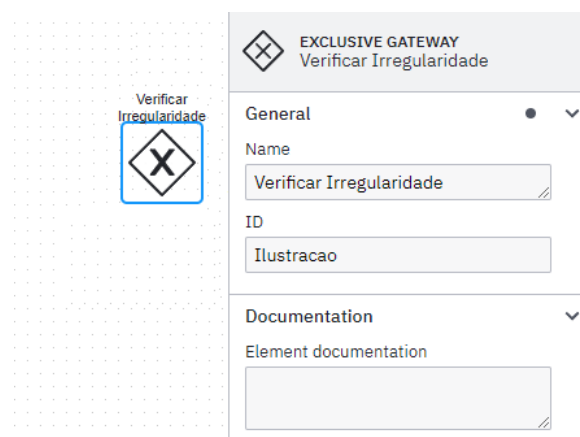
Figura 28 – User Task



Fonte: Elaborada pelo autor.

- *Exclusive Gateway*: Esse é o elemento representado por um "X" no centro, como indicado na figura 29. É empregado quando um processo alcança um ponto em que várias sequências são viáveis, mas apenas uma delas deve ser seguida. Ele representa uma decisão exclusiva, na qual a escolha do caminho a ser seguido depende de uma condição, conforme ilustrado no nosso BPMN. A representação desse elemento pode ser observada na figura 29.

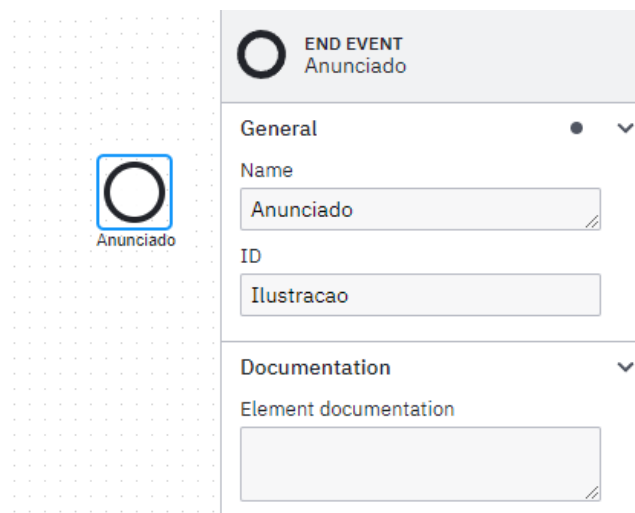
Figura 29 – Exclusive Gateway



Fonte: Elaborada pelo autor.

- *End Event*: Por fim, mas não menos importante, temos o evento final (*end event*), que é um elemento que simboliza o ponto de encerramento de uma instância do processo. Ele indica que o processo chegou ao seu término e que nenhuma outra atividade será executada após a ocorrência deste evento "*end event*". Este elemento é utilizado para marcar o encerramento do processo e a forma como os resultados são entregues. Ele é representado por um círculo com uma borda mais ampla, como pode ser observado na figura 30.

Figura 30 – End Event



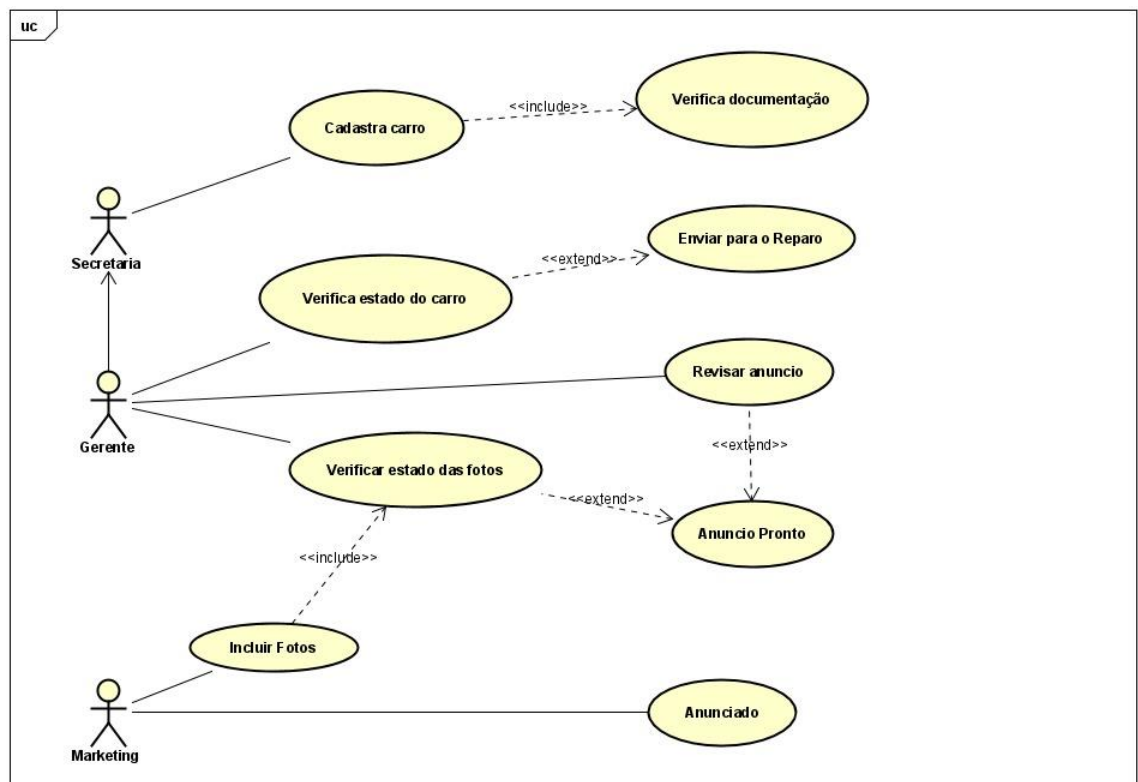
Fonte: Elaborada pelo autor.

Cada elemento "*User Task*" do nosso BPMN corresponde a uma interação com o usuário, ou seja, teremos uma tela correspondente para cada "*User Task*". Neste capítulo, vamos apresentar brevemente algumas dessas telas que estão associadas aos nossos processos.

4.3 Diagrama de casos de uso

Para modelar a lógica de processo em um sistema, um diagrama de atividades é uma ferramenta fundamental. Esse diagrama representa as atividades, decisões e interações, fornecendo uma visão clara da dinâmica do sistema. Ele é essencial para analisar e otimizar fluxos de trabalhos em projetos, a figura 31 a seguir mostra o nosso Diagrama de casos de uso.

Figura 31 – Diagrama de casos de uso



Fonte: Elaborada pelo autor.

4.4 Diagrama de classes

Através do Spring Boot, conseguimos criar e fazer referência a tabelas no banco de dados. Vamos ilustrar apenas uma das entidades presentes no nosso projeto, antes de apresentar o nosso diagrama de classes, com o objetivo de demonstrar a sua estrutura. Esta entidade corresponde a uma tabela no nosso banco de dados H2. Na figura 32 a seguir, temos uma demonstração dos dados dessa tabela, utilizando o Spring Boot.

Figura 32 – Referência de entidades

```
@Entity
@Table(name = "empresa_veiculo")
@Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
public class EmpresaVeiculo implements Serializable {

    no usages
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "sequenceGenerator")
    @SequenceGenerator(name = "sequenceGenerator")
    private Long id;

    3 usages
    @Column(name = "renavam")
    private String renavam;

    3 usages
    @Column(name = "placa")
    private String placa;

    3 usages
    @Column(name = "doc_regular")
    private Boolean docRegular;

    3 usages
    @Column(name = "multa", precision = 21, scale = 2)
    private BigDecimal multa;

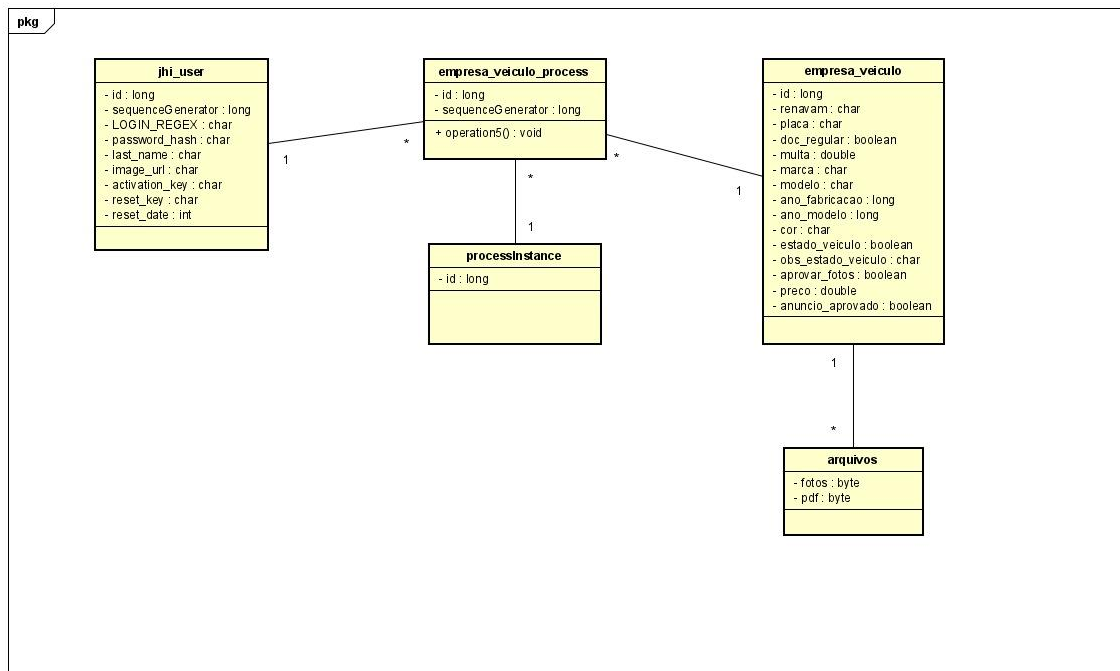
    3 usages
    @Column(name = "marca")
    private String marca;

    3 usages
    @Column(name = "modelo")
    private String modelo;
```

Fonte: Elaborada pelo autor

Como mencionado anteriormente, as entidades correspondem às tabelas dentro do banco de dados. A representação do nosso diagrama de classes apresenta os relacionamentos entre as tabelas que possuímos. Este diagrama exemplifica as classes, seus atributos, métodos e conexões, proporcionando uma visão abrangente da arquitetura do sistema. Conforme ilustrado na figura 33, temos o nosso diagrama de classes.

Figura 33 – Diagrama de Classes



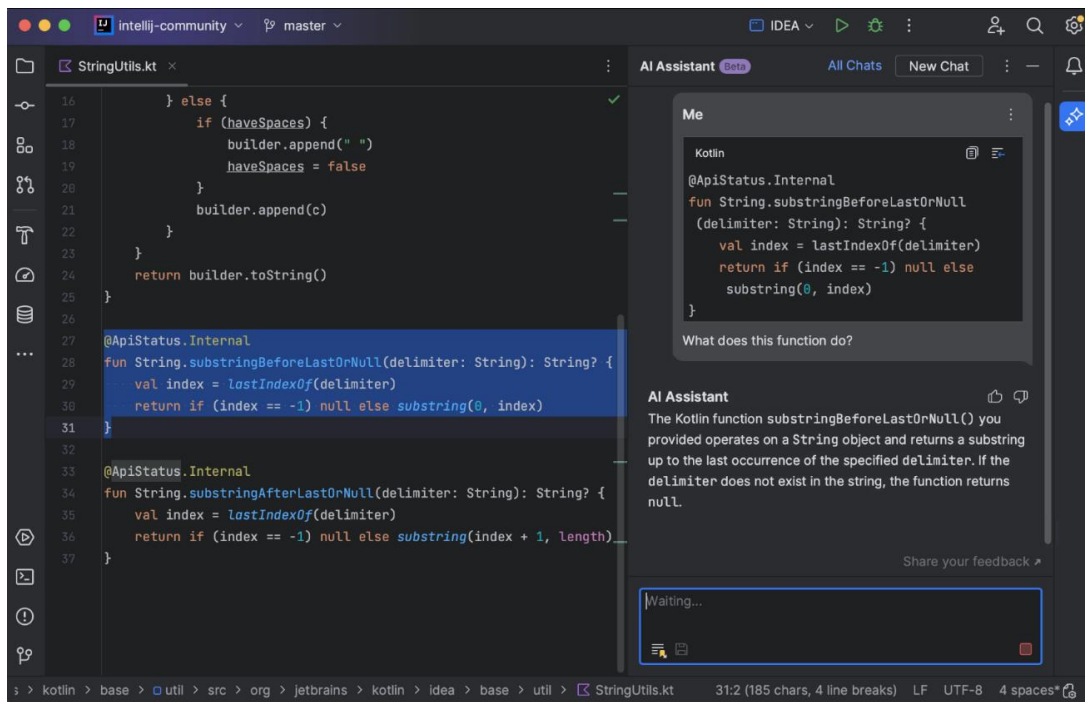
Fonte: Elaborada pelo autor.

4.5 Descrição da Implementação

Neste tópico, vamos exemplificar algumas das ferramentas que foram empregadas no projeto e oferecer uma breve visão de como elas foram implementadas na aplicação.

Antes de abordarmos a ferramenta JHipster, é importante introduzir a IDE (Ambiente de Desenvolvimento Integrado) que será utilizada. A IDE é simplesmente uma ferramenta de software que proporciona o ambiente de desenvolvimento. O IntelliJ IDEA, referido como IntelliJ, é uma das IDEs mais populares e amplamente adotadas. Desenvolvida pela JetBrains, ela é representada de forma concisa na figura 34, que fornece uma visão visual de sua interface.

Figura 34 - IntelliJ IDEA



Fonte: (INTELLIJ IDEA, 2023).

4.5.1 Desenvolvimento rápido com JHipster

Inicialmente, empregamos uma ferramenta de código aberto que integra as tecnologias Java e JavaScript para simplificar o desenvolvimento de aplicativos, conhecida como JHipster (Java Hipster). O JHipster reúne estruturas e ferramentas de desenvolvimento em um ambiente único, o que permite que os desenvolvedores gerem seus códigos-fonte. Essa ferramenta é notável por sua capacidade de criar projetos Spring Boot. Como representado na figura 35, utilizamos o JHipster para a geração da aplicação.

Figura 35 – Criação com JHipster

```
Welcome to JHipster v7.0.1
Application files will be generated in folder: /home/jhipster/app/tcc

Documentation for creating an application is at https://www.jhipster.tech/creating-an-app/
If you find JHipster useful, consider sponsoring the project at https://opencollective.com/generator-jhipster

JHipster update available: 8.0.0-rc.1 (current: 7.0.1)
Run npm install -g generator-jhipster to update.

Which *type* of application would you like to create? Monolithic application (recommended for simple projects)
What is the base name of your application? ProjectTCCUnicarioca
Info Using blueprint generator-jhipster-agilekip for server subgenerator
Info Using blueprint generator-jhipster-agilekip for client subgenerator
Do you want to make it reactive with Spring WebFlux? Yes
What is your default Java package name? tccproject
Which *type* of authentication would you like to use? JWT authentication (stateless, with a token)
Which *type* of database would you like to use? SQL (H2, PostgreSQL, MySQL, MariaDB, MSSQL)
Which *production* database would you like to use? MySQL
Which *development* database would you like to use? H2 with disk-based persistence
Would you like to use Maven or Gradle for building the backend? Maven
Do you want to use the JHipster Registry to configure, monitor and scale your application? Yes
Which other technologies would you like to use?
Which *Framework* would you like to use for the client? Vue
Do you want to generate the admin UI? Yes
Would you like to use a Bootswatch theme (https://bootswatch.com/)? Default JHipster
Would you like to enable internationalization support? Yes
Please choose the native language of the application Portuguese (Brazilian)
Please choose additional languages to install English
Besides JUnit and Jest, which testing frameworks would you like to use?
Would you like to install other generators from the JHipster Marketplace? (y/N) ☐
```

Fonte: Elaborada pelo autor.

Como representado na figura anterior, empregamos a ferramenta JHipster para criar a nossa aplicação. Durante a sua configuração, pudemos definir diversas estruturas, como o tipo de banco de dados a ser utilizado, conforme mencionado em capítulos anteriores, escolhemos o banco de dados H2, que inclui seus próprios códigos SQL. Além disso, determinamos um framework para o front-end através do JHipster, optando pelo uso do Vue.js.

No âmbito do JHipster, também optamos pela instalação do Docker. Em resumo, o Docker é uma plataforma de contêineres que possibilita que os desenvolvedores empacotem aplicativos e suas dependências em unidades isoladas conhecidas como “contêineres”. Esses contêineres podem ser executados de maneira consistente em qualquer ambiente compatível com o Docker, tornando-os mais eficientes e portáteis, ao mesmo tempo em que proporcionam isolamento e segurança. A figura 36 ilustra a instalação realizada via terminal (AGILEKIP, 2021).

Figura 36 – Instalação do Docker

```
HOST-MACHINE$ cd MY_WORK_DIRECTORY
HOST-MACHINE$ docker container run \
  --name agilekip.v0.0.12 \
  -v $PWD:/home/jhipster/app \
  -d -t agilekip/generator-jhipster-agilekip:v0.0.12
```

Fonte: Elaborada pelo autor.

4.5.2 Conexão com banco de dados

Primeiramente, vamos estabelecer a conexão com o banco de dados da nossa escolha, que é o Banco de Dados H2. A conexão é bastante simples, conforme ilustrado na figura 37, onde podemos observar como realizamos a conexão com o banco, utilizando o framework Spring Boot.

Figura 37 – Conexão com banco de dados

```
spring:
  devtools:
    restart:
      enabled: true
      additional-exclude: static/**,.h2.server.properties
    liveload:
      enabled: false # we use Webpack dev server + BrowserSync for liveload
  jackson:
    serialization:
      indent-output: true
  datasource:
    type: com.zaxxer.hikari.HikariDataSource
    url: jdbc:h2:file:./target/h2db/db/newproject;DB_CLOSE_DELAY=-1
    username: newProject
    password:
    hikari:
      poolName: Hikari
      auto-commit: false
  h2:
    console:
      # disable spring boot built-in h2-console since we start it manually with correct configuration
      enabled: false
  jpa:
    database-platform: tech.jhipster.domain.util.FixedH2Dialect
```

Fonte: Elaborada pelo autor.

Após a conexão com o banco de dados H2, estaremos prontos para utilizá-lo. No entanto, antes disso, precisamos configurar a conexão com a “PORTA”. A figura 38 a seguir ilustra como essa conexão foi estabelecida.

Figura 38 – Conexão de Porta

```
private final Environment env;

+
public DatabaseConfiguration(Environment env) { this.env = env; }

/**
 * Open the TCP port for the H2 database, so it is available remotely.
 *
 * @return the H2 database TCP server.
 * @throws SQLException if the server failed to start.
 */
+
+
@Bean(initMethod = "start", destroyMethod = "stop")
@Profile(JHipsterConstants.SPRING_PROFILE_DEVELOPMENT)
public Object h2TCPServer() throws SQLException {
    String port = getValidPortForH2();
    log.debug("H2 database is available on port {}", port);
    return H2ConfigurationHelper.createServer(port);
}
```

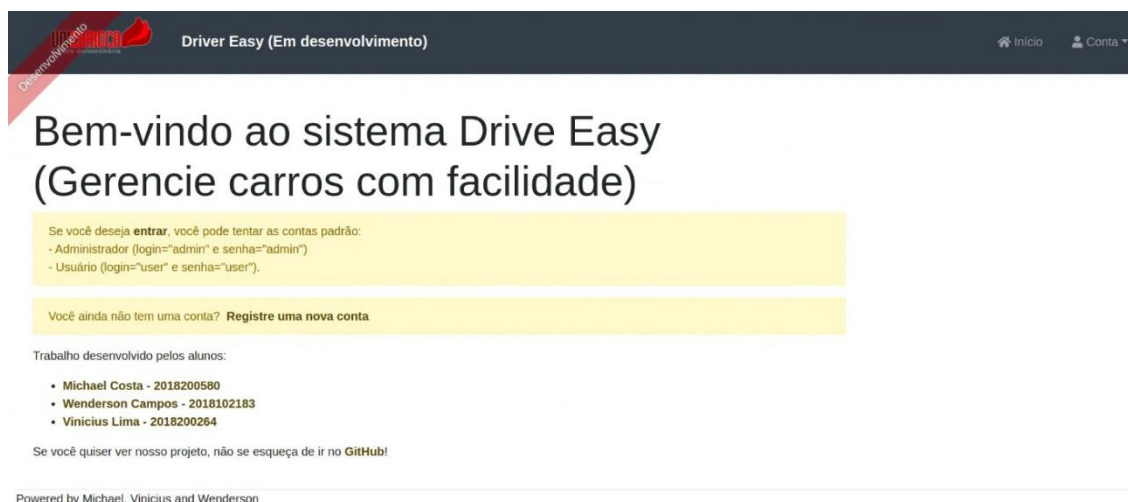
Fonte: Elaborada pelo autor.

4.6 Interfaces com o Usuário

O nosso objetivo neste projeto é destacar a eficiência e segurança da nossa aplicação. Vamos utilizar o Camunda em conjunto com uma BPMN, o que requer configurações específicas dentro da aplicação, incluindo a configuração das *User Task* e *Service Task*.

Para começar, apresentaremos a nossa tela inicial. Todas as telas foram desenvolvidas com o auxílio de dois frameworks, Bootstrap e, principalmente, Vue.js. Nesta tela, os usuários têm a opção de se cadastrar ou fazer login como um usuário existente. A figura 39 ilustra essa demonstração, na qual exibimos uma tela de registro que ainda não está vinculada a nenhum processo no BPMN, mas permite o cadastro dos usuários.

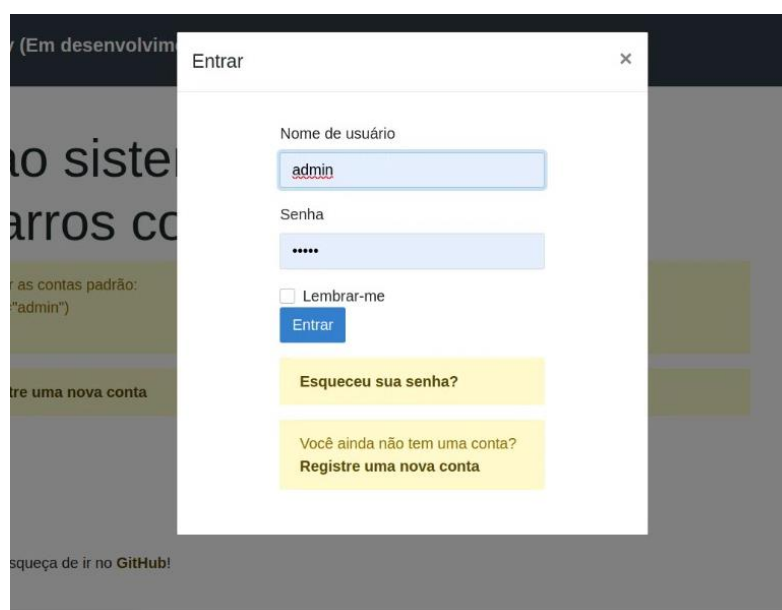
Figura 39 – Tela inicial



Fonte: Elaborada pelo autor.

Ao acessar a tela de login, entramos na interface onde é possível efetuar o processo de login. Nesta tela, são apresentadas opções adicionais, como a possibilidade de lembrar o nome de usuário e a senha. Por padrão, criamos um login de administrador, que tem acesso a todas as funções do sistema. Além disso, disponibilizamos a opção “Esqueceu sua senha?” que permite ao sistema enviar um e-mail ao usuário como meio de recuperação da conta. Na figura 40, ilustramos a tela de login.

Figura 40 – Tela de login



Fonte: Elaborada pelo autor.

4.6.1 Protegendo senhas com JWT

Aqui, fazemos uso de uma das ferramentas disponíveis no Spring, o Spring Security, que proporciona segurança na transferência de dados para o nosso banco de dados H2. Com o Spring Security, utilizamos JWTs (*Token de Web JSON*) para autenticar os usuários e autorizá-los com base nas informações contidas nos tokens. No sistema, implementamos uma configuração de segurança que restringe o acesso à API apenas a partir de autenticação por Token (SILVEIRA, FRANCIS, 2019).

Os dados de login são transmitidos de forma completamente segura para o nosso banco de dados, sendo a senha armazenada de maneira totalmente criptografada. Como ilustrado na figura 41 a seguir, apresentamos uma representação de como esses dados são inseridos no nosso banco de dados.

Figura 41 – Dados Criptografados JWT



The image shows a screenshot of a database query result in a web application. At the top, the SQL query 'SELECT * FROM JHI_USER;' is displayed. Below it is a table with 6 columns: ID, LOGIN, PASSWORD_HASH, FIRST_NAME, LAST_NAME, and EMAIL. There are two rows of data. The first row represents an 'admin' user with a long, complex password hash, first name 'Administrator', last name 'Administrator', and email 'admin@localhost'. The second row represents a 'user' user with a similar password hash, first name 'User', last name 'User', and email 'user@localhost'. Below the table, it says '(2 rows, 5 ms)'. At the bottom left of the table area is an 'Edit' button.

ID	LOGIN	PASSWORD_HASH	FIRST_NAME	LAST_NAME	EMAIL
1	admin	\$2a\$10\$gSAhZrxMlrbgj/kkK9UceBPpChGWJA7SYlb1Mqo.n5aNLq1/oRrC	Administrator	Administrator	admin@localhost
2	user	\$2a\$10\$VEjxo0jq2YG9Rbk2HmX9S.k1uZBGYUhdUcid3g/vfiEI7lwWgOH/K	User	User	user@localhost

(2 rows, 5 ms)

Edit

Fonte: Elaborada pelo autor.

Agora, passaremos para a criação de um registro, onde existem campos que precisam ser preenchidos corretamente para concluir o processo de registro. Esses campos incluem nome, e-mail, senha e confirmação de senha, com formatos específicos necessários. Na figura 42, apresentamos essa demonstração, na qual preenchemos incorretamente o registro. Como resultado, a aplicação não avançará para a próxima tela.

Figura 42 – Tela do cadastro de usuário

Registro

Nome de usuário

Seu nome de usuário é obrigatório.

E-mail

Seu e-mail é inválido.
Seu e-mail deve ter pelo menos 5 caracteres.

Nova senha

Sua senha deve ter pelo menos 4 caracteres.

Confirmação da nova senha

Sua senha de confirmação deve ter pelo menos 4 caracteres.
A senha e sua confirmação não correspondem!

Registrar

Fonte: Elaborada pelo autor.

Agora, vamos acessar a conta de administrador, que tem acesso a todos os recursos e funcionalidades do sistema. Após efetuarmos o login como administrador, o sistema nos desbloqueará funcionalidades adicionais. Na barra de navegação, teremos acesso a recursos administrativos que exibirão todos os usuários cadastrados até o momento. Com as funcionalidades administrativas, poderemos criar novos usuários, visualizar, editar, excluir e também ajustar algumas de suas permissões, tudo de maneira totalmente segura e funcional. Conforme ilustrado figura 43, onde é exibido essas informações.

Figura 43 – Exibição de usuários

Driver Easy (Em desenvolvimento)

[Início](#) [Minhas Tarefas](#) [Entidades](#) [Administração](#) [Conta](#)

Usuários

[Atualizar lista](#) [+ Criar um novo usuário](#)

ID	Login	E-mail	Idioma	Perfis	Data de criação	Modificado por	Data de modificação	
1	admin	admin@localhost	Ativado	en	ROLE_USER ROLE_ADMIN	system		Visualizar Editar Excluir
2	user	user@localhost	Ativado	en	ROLE_USER	system		Visualizar Editar Excluir

Mostrando 1 - 2 de 2 itens.

[«](#) [<](#) [1](#) [>](#) [»](#)

Powered by Michael, Vinicius and Wenderson

Fonte: Elaborada pelo autor.

Na figura 43 apresentada anteriormente, temos a opção de editar um usuário. Vamos selecioná-la e demonstrar como esse processo é eficiente. Tanto na edição quanto na criação de um novo usuário por meio das funcionalidades de um administrador, temos a capacidade de definir suas permissões. Na figura 44 a seguir, apresentamos a funcionalidade exclusiva de um administrador, que permite até mesmo alterar as permissões de um usuário no sistema. No campo de “perfis”, podemos configurar suas permissões dentro do sistema.

Figura 44 – Editar usuário

O formulário, intitulado "Criar ou editar um usuário", contém os seguintes elementos:

- Login:** Um campo de texto com uma barra vermelha vertical à esquerda.
- Nome:** Um campo de texto com o placeholder "Seu primeiro nome".
- Sobrenome:** Um campo de texto com o placeholder "Seu sobrenome".
- E-mail:** Um campo de texto com o placeholder "Seu e-mail" e uma barra vermelha vertical à esquerda.
- Ativado:** Uma caixa de seleção desativada (☐) com o rótulo "Ativado".
- Idioma:** Um menu suspenso com uma seta para baixo.
- Perfis:** Um menu suspenso com uma barra deslizante, mostrando as opções "ROLE_ADMIN" e "ROLE_USER".
- Botões:** "Cancelar" (com ícone de X) e "Salvar" (com ícone de disco).

Fonte: Elaborada pelo autor.

4.6.2 Interface para BPMN

Agora, adentrando na seção em que mencionamos o Camunda BPMN, que é o cerne da nossa aplicação, temos uma ferramenta que nos permite modelar processos na nossa aplicação. Em qualquer sistema que envolva a utilização do Camunda, isso resulta em melhorias significativas na eficiência. O Camunda possibilita a automação de fluxos de trabalho e processos de negócios, o que reduz a necessidade de tarefas manuais e acelera o tempo de conclusão das atividades.

Além disso, contamos com a flexibilidade e escalabilidade do sistema, o que significa que ao longo do uso do sistema, podemos adicionar novos processos. Isso facilita a manutenção e permite a inclusão de novos processos.

Vamos agora abordar o uso desse recurso. Na área de administração, temos a capacidade de realizar o *deploy* do nosso BPMN, que já foi apresentado anteriormente na figura 25. Em resumo, na programação, o termo “*deploy*” refere-se à ação de tornar-se um aplicativo ou sistema de software disponível e funcional para uso. A figura 45 ilustra como podemos realizar o *deploy* do nosso BPMN.

Figura 45 – Deploy BPMN



Fonte: Elaborada pelo Autor.

Após realizar o *deploy* do nosso BPMN, avançaremos em cada processo que foi criado, começando pelo *Start Form*. Como mencionado anteriormente, nesse ponto, inseriremos os dados iniciais dentro do nosso *Start Form*.

No início, forneceremos o Renavam e a placa de um veículo, que são informações iniciais. Primeiramente, cadastraremos essas informações do veículo no sistema. É importante lembrar que este é o primeiro processo ao qual temos acesso dentro do BPMN, sendo ele o *Start Form*. A figura 46 é ilustrado esse campo preenchido com as informações.

Figura 46 – Tela Start Form

Criar ou editar um Formulário Inicial da Empresa de Veículo

#1001 - EmpresaVeiculoProcess

Somos especializados na venda de veículos novos e usados, nacionais e importados. Com certeza você não só apreciará como irá comprar seu veículo conosco. Todos nossos veículos são revisados criteriosamente, possibilitando dar aos nossos clientes tranquilidade na hora da compra. Não perca tempo! Compre seu veículo com quem mais entende do assunto. Nossos vendedores terão o prazer em atendê-lo.

Renavam

111111111111111111

Placa

TVV-0003

BPMN Process Definition Id : EmpresaVeiculoProcess

Cancelar Start

Fonte: Elaborada pelo autor

Após passarmos pelo primeiro processo, avançaremos para uma *Service Task*, onde se encontra uma regra de negócio. Em Java, chamamos isso de *Delegate*, onde definiremos a ação correspondente.

Este *Delegate* terá a responsabilidade de determinar de forma aleatória se o documento do veículo está regular ou não. Observe que existe um *Exclusive Gateway* que representa uma condição. Caso o documento do veículo esteja regular, o próximo processo será “Inclusão de Documento”. No entanto, se, de forma aleatória, o documento estiver irregular, o processo seguinte será “Regularização de Veículo”. Na estruturação, definimos que o fator que torna o documento irregular é uma multa.

Antes de abordarmos a próxima “Tarefa de Usuário”, vamos primeiro explorar a *Service Task*. Na figura 47 ilustra esse *Service Task* que determinará se o documento está regular ou não.

Figura 47 – Delegate de Verificação de Documento

```
package com.mycompany.myapp.process.empresaveiculoProcess.delegate;

import com.mycompany.myapp.domain.EmpresaVeiculo;
import com.mycompany.myapp.service.EmpresaVeiculoService;
import com.mycompany.myapp.service.dto.EmpresaVeiculoDTO;
import java.util.Optional;
import java.util.Random;
import org.camunda.bpm.engine.delegate.DelegateExecution;
import org.camunda.bpm.engine.delegate.JavaDelegate;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class VerificarDocDelegate implements JavaDelegate {

    // usage
    private final Logger log = LoggerFactory.getLogger(VerificarDocDelegate.class);

    @Autowired
    private EmpresaVeiculoService empresaVeiculoService;

    @Override
    public void execute(DelegateExecution delegateExecution) throws Exception {
        String businessKey = delegateExecution.getBusinessKey();
        String idVeiculo[] = businessKey.split("regex:");

        Optional<EmpresaVeiculoDTO> empresaVeiculoDTO = empresaVeiculoService.findOne(Long.valueOf(idVeiculo[1]));

        if (empresaVeiculoDTO.isPresent()) {
            Random random = new Random();
            empresaVeiculoDTO.get().setDocRegular(random.nextBoolean());
            empresaVeiculoService.save(empresaVeiculoDTO.get());
        } else {
            log.info("Veiculo nao encontrado no VerificarDocDelegate");
        }
    }
}
```

Fonte: Elaborada pelo autor.

De maneira aleatória, foi determinado que o documento não está regular, o que significa que avançaremos para o processo “Regularização de Veículo”. Neste processo, precisamos regularizar o veículo, o que inclui o pagamento de uma multa. Posteriormente, um usuário pode efetuar o pagamento da multa e regularizar o veículo. É importante ressaltar que não é possível avançar para o próximo processo a menos que o documento esteja regular. A figura 48 nos apresenta como esse processo funciona.

Figura 48 – Tela regularizar veículo.

Renavam
1111111111111111

Placa
TVV-0003

Doc Regular
☒

Multa
10000

Created at: Tuesday, October 31, 2023 at 9:39 PM

[← Voltar](#) [Complete](#)

Fonte: Elaborada pelo autor.

Após concluirmos esse processo, prosseguiremos para a próxima “Tarefa de Usuário”, que diz respeito à inclusão dos documentos restantes do veículo. Nesse estágio, entramos no processo de “Inclusão de Documentos”, no qual já encontramos os campos preenchidos com as informações dos processos anteriores.

Neste processo, surgirão novos campos que preencheremos com as demais informações referentes ao veículo. A figura 49 nos mostra a interface deste processo.

Figura 49 – Tela Inclusão de Documento

Renavam
111111111111111111

Placa
TVV-0003

Doc Regular
☒

Multa

Marca
Volkswagen

Modelo
Gol

Ano Fabricacao
2020

Ano Modelo
2021

Cor
Branco

Created at: Tuesday, October 31, 2023 at 9:35 PM

Fonte: Elaborada pelo autor.

Após concluirmos esse processo, criamos um cenário no qual alguém verificará o estado físico do veículo e emitirá uma avaliação sobre o seu estado. Neste cenário, estabelecemos que o sistema tomará uma decisão aleatória sobre o estado do veículo. Por exemplo, se o veículo tiver a lataria amassada, uma observação será registrada para justificar por que o veículo não está apto para cadastro. Como resultado, ele não prosseguirá para a próxima “Tarefa de Usuário”. Nesse ponto, teremos uma condição que nos levará ao *Exclusive Gateway*, que acionará o próximo processo com base na resposta.

No exemplo que construímos mentalmente, o veículo não está apto para cadastro devido a um defeito na lataria, e essa observação será registrada no processo “Verificar Estado do Veículo”. A figura 50 ilustra essa tela.

Figura 50 – Tela verificar estado do veículo.

The screenshot shows a web form with the following elements:

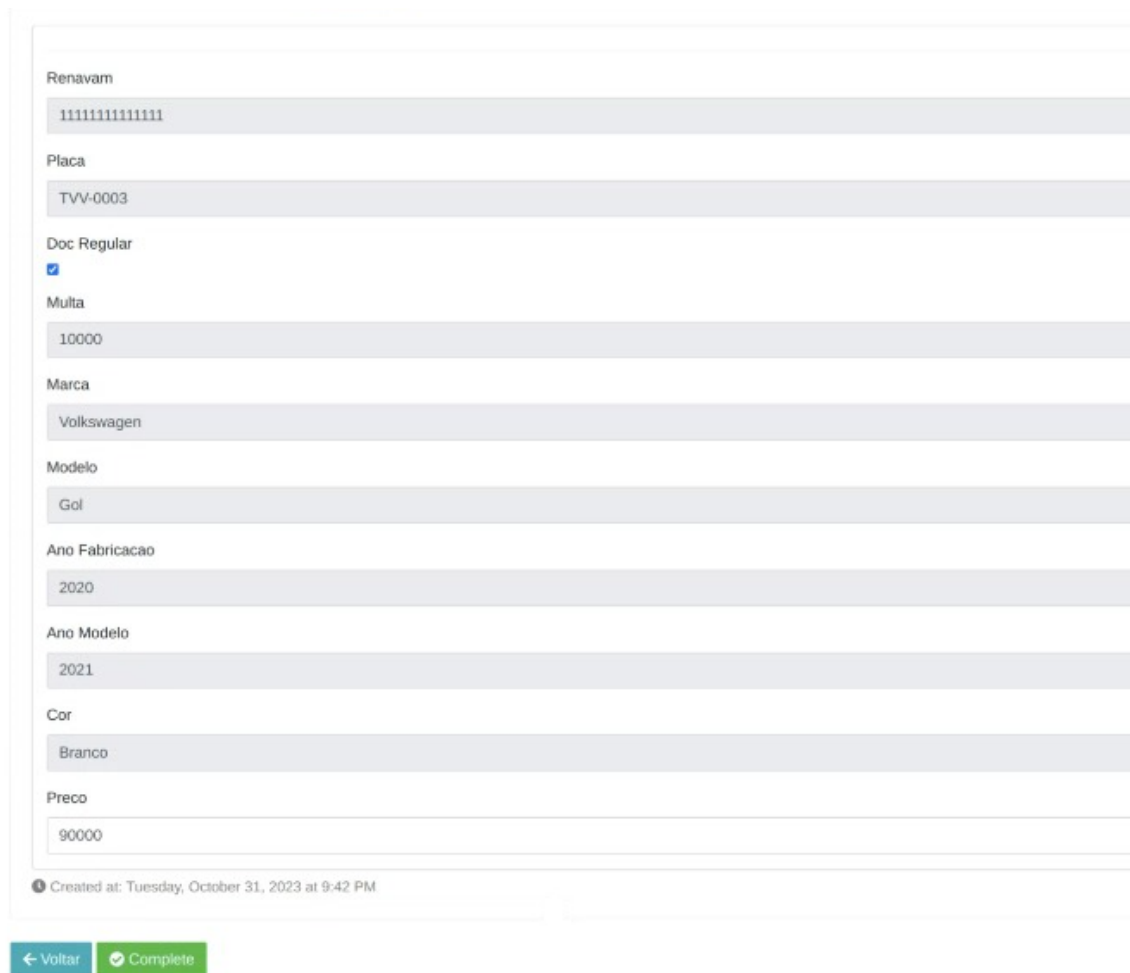
- Estado Veiculo**: A label above a checkbox.
- Obs Estado Veiculo**: A label above a text input field containing the text "Lataria amassada".
- Created at**: A timestamp "Tuesday, October 31, 2023 at 9:37 PM" preceded by a clock icon.
- Buttons**: Two buttons at the bottom, "Voltar" (blue with a left arrow) and "Complete" (green with a checkmark).
- Footer**: The text "is and Wenderson" at the very bottom.

Fonte: Elaborada pelo autor.

Assim, avançaremos para o nosso segundo *Service Task* do BPMN, no qual definiremos o estado do veículo. Toda essa ação será realizada por meio de um *Delegate*.

Agora, chegamos ao processo que envolve o cadastro do veículo no sistema. Aqui, será necessário preencher apenas o campo referente ao valor do veículo que será inserido no sistema. No processo “Cadastrar Anúncio”, estaremos inserindo um valor, como ilustrado na figura 51.

Figura 51 – Tela cadastrar anuncio



Formulário de cadastro de anúncio com os seguintes campos:

- Renavam: 111111111111111111
- Placa: TVV-0003
- Doc Regular: ☒
- Multa: 10000
- Marca: Volkswagen
- Modelo: Gol
- Ano Fabricacao: 2020
- Ano Modelo: 2021
- Cor: Branco
- Preco: 90000

Created at: Tuesday, October 31, 2023 at 9:42 PM

Botões de ação: Voltar, Complete

Fonte: Elaborada pelo autor.

Como último passo na interação do usuário, temos a revisão do anúncio por parte do gerente, que é responsável por avaliar se o anúncio está de acordo com a visão do gerente. Além disso, o gerente pode identificar erros nos preços ou informações inseridas incorretamente. Em caso de reprovação, o cadastro retornaria ao processo anterior para as devidas correções, reiniciando o processo de revisão. Esse ciclo continua até que o usuário aprove o cadastro, encerrando o processo.

A tela de revisão exibe as informações do cadastro, e o gerente terá a opção de simplesmente aprovar ou não o anúncio. A figura 52 ilustra esse processo.

Figura 52 – Tela revisão de cadastro.

The screenshot displays a registration review form with the following fields and values:

- Placa:** TVV-0003
- Doc Regular:** ☒
- Multa:** (empty field)
- Marca:** GM
- Modelo:** Onix
- Ano Fabricacao:** 2023
- Ano Modelo:** 2023
- Cor:** Cinza
- Preco:** 90000
- Anuncio Aprovado:** ☒

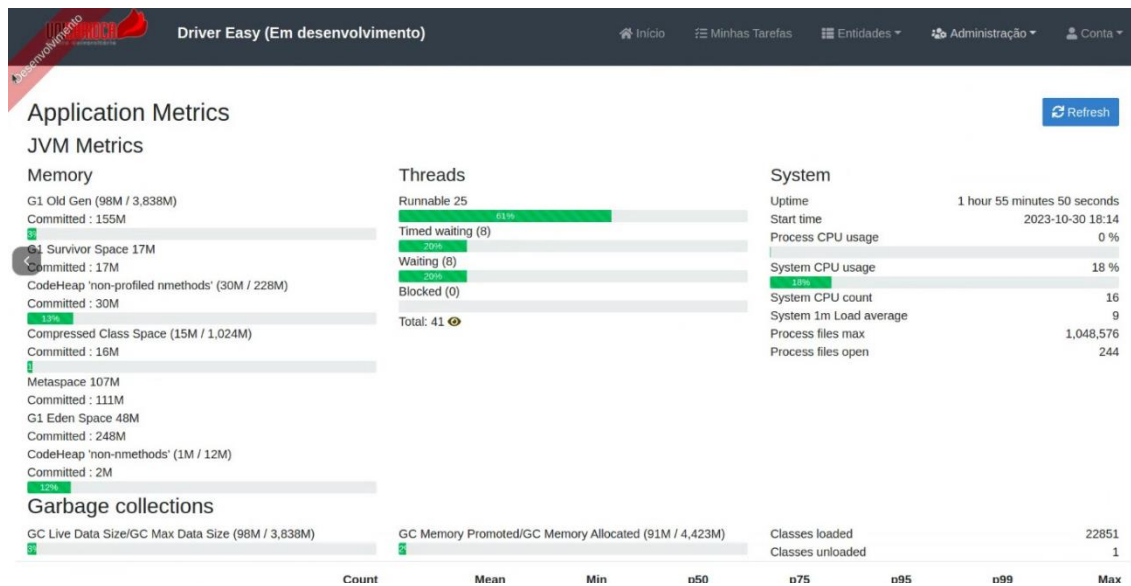
At the bottom of the form, it states: "Created at: Tuesday, October 31, 2023 at 9:45 PM". Below the form, there are two buttons: "Voltar" (Back) and "Complete".

Fonte: Elaborada pelo autor.

4.7 Métricas de aplicação

Por último, gostaríamos de destacar a eficiência da aplicação, incluindo uma das funcionalidades administrativas: as métricas da aplicação. Nesse recurso, encontramos informações detalhadas sobre o tempo de resposta, o consumo de memória e outros dados relevantes. A figura 53 apresentamos essas métricas.

Figura 53 – JVM Metrics



Fonte: Elaborada pelo autor.

5 CONCLUSÃO

A gestão de estoque é um processo de grande importância nas organizações que fazem uso de estoque em suas operações, e essa importância se torna evidente ao examinar o contexto de uma concessionária. A gestão eficaz do estoque em uma concessionária é crucial para manter o equilíbrio financeiro e atender às demandas dos clientes. Este projeto visa fornecer uma plataforma que simplifica o processo de gerenciamento de estoque, proporcionando visibilidade, automação de processos e otimização das operações.

A partir do modo de operação e das necessidades dos usuários, foram definidos os requisitos funcionais e não funcionais que serviram de base para a construção do sistema. A partir desses requisitos, foram desenvolvidos diversos diagramas que compõem este trabalho, incluindo o diagrama de casos de uso, que identificou os atores que interagirão com o sistema. O diagrama de atividades, por sua vez, proporcionou uma representação visual do fluxo de atividades envolvidas na geração de contratos, e o modelo ER (Entidade-Relacionamento) permitiu a visualização das entidades persistidas no sistema.

A escolha das tecnologias para o desenvolvimento foi baseada em uma análise cuidadosa, incluindo a revisão de trabalhos relacionados e a investigação de soluções já existentes no mercado. Optou-se por utilizar tecnologias amplamente adotadas e com comunidades robustas, sendo o framework Vue.js para o desenvolvimento do front-end, o Spring Boot para o desenvolvimento do back-end e a máquina de processos Camunda, que oferece uma plataforma tecnologicamente avançada e eficaz para o gerenciamento de estoque.

A adoção da plataforma Camunda, juntamente com a notação BPMN (*Business Process Model and Notation*), permitiu a modelagem e a automação eficiente dos fluxos de trabalho, resultando na melhoria da eficiência das operações. A escolha criteriosa das tecnologias permitiu a definição dos requisitos mínimos necessários para a implementação bem-sucedida do sistema em um ambiente local e dedicado.

Expressamos nossa gratidão a todas as pessoas que contribuíram para o sucesso deste projeto, incluindo o nosso orientador, colegas e familiares, cujo apoio foi inestimável.

Em conclusão, este TCC demonstra a viabilidade e o valor de uma aplicação Vue.js com Spring Boot e a máquina de processos Camunda no gerenciamento de estoque de uma concessionária. Esperamos que este trabalho inspire pesquisas adicionais e aplicações

práticas nesse campo. O sucesso deste projeto ressalta a importância da inovação tecnológica e da automação de processos para melhorar a eficiência operacional e atender às necessidades do mercado.

REFERÊNCIAS

- AGILEKIP. **Installing AgileKIP Generator**. AGILEKIP. 2021. Disponível em <<https://agilekip.github.io/pap-documentation/tutorials/installation>>. Acesso em 17 outubro de 2023.
- AWARI. **Aprenda Desenvolvimento Web Front-End Interativo Com Javascript E JQuery**. Awari. 2023. Disponível em <<https://awari.com.br/aprenda-desenvolvimento-web-front-end-interativo-com-javascript-e-jquery/>>. Acesso em 8 setembro de 2023.
- BOOTSTRAP 2023 – HOSTINGER
BRASIL. RICARDO. HOSTINGER, 2020 (11min). Disponível em <https://www.youtube.com/watch?v=jsTJL6Da5wc&ab_channel=HostingerBrasil> Acesso em 28 setembro de 2023.
- CAMUNDA Modeler. Camunda. 2023. Disponível em <<https://camunda.com/download/modeler/>>. Acesso em 13 outubro de 2023.
- CARVALHO, LUANA. **HTML5 é finalmente lançado na íntegra**. Impacta. 2014. Disponível em <<https://www.impacta.com.br/blog/html5-e-finalmente-lancado-na-integra/>>. Acesso em 12 setembro de 2023.
- CRONAPP. 2019. Disponível em <<https://blog.cronapp.io/frameworks-para-desenvolvimento-de-sofwares/>>. Acesso em 26 setembro de 2023.
- ECOSISTEMA Spring. Ikastaroak. 2023. Disponível em <https://ikastaroak.ulhi.net/edu/es/DAW/DWES/DWES07/es_DAW_DWES07_Conte_nidos/website_1221_ecosistema_spring.html>. Acesso em 1 outubro de 2023.
- FADATARE, RAMESH. **Spring Stereotype Annotations**. Java Guides. 2023. Disponível em <<https://www.javaguides.net/2023/01/spring-boot-component-controller.html>>. Acesso em 7 outubro de 2023.
- GETBOOTSTRAP. 2023^a. Disponível em <<https://getbootstrap.com/docs/4.0/about/history/>>. Acesso em 26 setembro de 2023.
- GETBOOTSTRAP. 2023b. Disponível em <<https://getbootstrap.com/>>. Acesso em 26 setembro de 2023.
- GETBOOTSTRAP. 2023c. Disponível em <<https://getbootstrap.com/docs/5.3/getting-started/introduction/>>. Acesso em 29 setembro de 2023.
- GETBOOTSTRAP. 2023d. Disponível em <<https://getbootstrap.com/docs/4.0/components/buttons/>>. Acesso em 29 outubro de 2023.
- GETBOOTSTRAP. 2023e. Disponível em <<https://getbootstrap.com/docs/4.1/getting-started/introduction/>>. Acesso em 29 setembro de 2023.
- GIAMONIANO, IGOR. **Direto ao ponto: O que é CRUD?**. Medium. 2020. Disponível em <<https://igorgiamoniano.medium.com/direto-ao-ponto-o-que-%C3%A9-crud-2c41b48fafee>>. Acesso em 16 setembro de 2023.

GUEDES, MARYLENE. **O que é o Vue.js?**. Treina WEB. 2020. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-o-vue-js>>. Acesso em 1 outubro de 2023.

H2(DATABASE). WIKIPÉDIA. 2023. Disponível em <[https://en.wikipedia.org/wiki/H2_\(database\)](https://en.wikipedia.org/wiki/H2_(database))>. Acesso em 7 outubro de 2023.

IBMN. **O que é Java Spring Boot?**. IBM. 2023. Disponível em <<https://www.ibm.com/br-pt/topics/java-spring-boot>>. Acesso em 6 outubro de 2023.

INTELLIJ IDEA. JetBrains. 2023. Disponível em <<https://www.jetbrains.com/pt-br/idea/>>. Acesso em 16 outubro de 2023.

JAYE. **How HTML, CSS, and JavaScript work**. Scrimba. 2022. Disponível em <<https://scrimba.com/articles/html-css-javascript/>>. Acesso em 11 setembro de 2023.

KONDADO. **Banco de Dados: O que é e quais são os principais tipos?**. KONDADO. 2022. Disponível em <<https://kondado.com.br/blog/blog/2022/09/13/banco-de-dados-o-que-e-e-quais-sao-os-principais-tipos/>>. Acesso em 15 setembro de 2023.

KOTLER. **Philip Kotler: história, contribuições e influência do maior nome do Marketing**. Rockcontent. 2018. Disponível em <<https://rockcontent.com/br/blog/philip-kotler/>>. Acesso em 1 setembro de 2023.

LOPES, MICHELE. **O que é um Desenvolvedor Web Front-End e o que ele faz**. Ebac. 2023. Disponível em <<https://ebaconline.com.br/blog/desenvolvedor-front-end-o-que-faz>>. Acesso em 12 setembro de 2023.

MADEIRA, MARCELO. **JVM, Portabilidade e Desempenho**. Celodemelo. 2007. Disponível em <<https://celodemelo.wordpress.com/2007/10/11/jvm-portabilidade-e-desempenho/>>. Acesso em 4 outubro de 2023.

MARTINS, MATEUS. **MVC vs MVVM – Diferença entre eles**. Guru99. 2023. Disponível em <<https://www.guru99.com/mvc-vs-mvvm.html>>. Acesso em 3 outubro de 2023.

MEDEIROS, RODRIGO. **Visualização de dados, UX e front-end: comunidades e conceitos que se encontram novamente**. Medium. 2017. Disponível em <<https://brasil.uxdesign.cc/visualiza%C3%A7%C3%A3o-de-dados-ux-e-front-end-comunidades-e-conceitos-que-se-encontram-novamente-755bb3a02ead>>. Acesso em 8 setembro de 2023.

MOZILLA. 2023. Disponível em <https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction#dom_e_javascript>. Acesso em 13 setembro de 2023.

OLIVEIRA, LUCAS. **O que É Backend e Frontend: Guia Completo para Iniciantes**. Awari. 2023. Disponível em <<https://awari.com.br/o-que-e-backend-e-frontend-guia-completo-para-iniciantes-3/#:~>>. Acesso em 6 setembro de 2023.

OLIVEIRA, LUCAS. **O que É Backend e Frontend: Guia Completo para Iniciantes**. Awari. 2023. Disponível em <<https://awari.com.br/o-que-e-backend-e-frontend-guia-completo-para-iniciantes-3/>>. Acesso em 15 setembro de 2023.

PATEL, BINAL. **Bootstrap vs Angular: Comparing The Two Ideal Frontend Frameworks**. Bacany Technology. 2023. Disponível em <<https://www.bacanytechnology.com/blog/bootstrap-vs-angular>>. Acesso em 28 de 2023.

QUITUMBA FERREIRA, Introdução ao Portlet. (24min). Disponível em<https://www.youtube.com/watch?v=JerxIX7b6PI&ab_channel=QuitumbaFerreira>. Acesso em 5 outubro de 2023.

RAMOS, LARISSA. **Front-end e suas ferramentas para profissionais de UI/UX**. Medium. 2023. Disponível em <<https://medium.com/@larimesira/front-end-e-suas-ferramentas-para-profissionais-de-ui-ux-db7ffffbe0e8>>. Acesso em 7 setembro de 2023.

SILVA, FLAVIO. **Vue.js**. 2023. 31 slides. Disponível em <<https://www.facom.ufu.br/~flavio/pi/files/2022-01/04-VUE.pdf>>. Acesso em 2 outubro de 2023.

SILVA, GIZELE. **O que é arquitetura MVVM?**. Coodesh. 2023. Disponível em <<https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvvm/>>. Acesso em 3 outubro de 2023.

SILVA, GIZELE. **O que é CRUD?**. Coodesh. 2023. Disponível em <<https://coodesh.com/blog/dicionario/o-que-e-crud/>>. Acesso em 16 setembro de 2023.

SILVERA, FRANCIS. **JWT: Prós e contras da sua utilização..** Objective. 2019. Disponível em <<https://www.objective.com.br/insights/jwt/>>. Acesso em 18 outubro de 2023.

SOUZA, Wesley. **Desenvolvimento front-end: como começar e principais tecnologias**. Zup. Disponível em <<https://www.zup.com.br/blog/desenvolvimento-front-end>>. Acesso em 15 setembro de 2023.

SPRING FRAMEWORK. DOCS SPRING. 2023. Disponível em <<https://docs.spring.io/spring-framework/docs/3.0.x/spring-framework-reference/html/overview.html>>. Acesso em 4 outubro de 2023.

SPRING FRAMEWORK. DOCS SPRING. 2023. Disponível em <<https://docs.spring.io/spring-framework/docs/3.0.x/spring-framework-reference/html/overview.html>>. Acesso em 5 outubro de 2023.

SUPREMA. Suprema Propaganda. 2016. Disponível em <<https://supremapropaganda.com.br/voce-sabe-como-funciona-um-site/exemplo-carro-back-end-front-end/>>. Acesso em 17 setembro de 2023.

THE UNIVERSAL PROCESS. CAMUNDA (2min). Disponível em <<https://youtu.be/ahL4JAQMDIM>>. Acesso em 8 outubro de 2023.

TWS. BENEFITS Of Choosing VueJS Development Services. 2023. Disponível em <<https://www.tekkiwebsolutions.com/vuejs-development-services/>>. Acesso em 2 outubro de 2023.

VUE.JS. **Aprofundando-se na Reatividade**. 2020. Disponível em <<https://vueframework.com/docs/v3/br/guide/reactivity.html#o-que-e-reatividade>>. Acesso em 1 outubro de 2023.

VUE.JS. **Getting Started**. VUE.JS. 2023. Disponível em <<https://012.vuejs.org/guide/>>. Acesso em 3 outubro de 2023.

WIKIPÉDIA. **Java (linguagem de programação)**. 2023. Disponível em <[https://pt.wikipedia.org/wiki/Java_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o))>. Acesso em 4 outubro de 2023.

ZARDO, ROSILENE. **Camunda**. LINKEDIN. 2023. Disponível em <<https://www.linkedin.com/pulse/camunda-rosilene-zardo/?originalSubdomain=pt>>. Acesso em 6 outubro de 2023.