

UNIVERSIDADE PAULISTA

JOÃO CAMARGO

PIM I Projeto Integrado Multidisciplinar

ARARAQUARA

2019

PIM III – Projeto Integrado Multidisciplinar

TRABALHO PARA A OBTENÇÃO DO TÍTULO
DE TECNÓLOGO EM
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
APRESENTADO A
UNIVERSIDADE PAULISTA
ORIENTADO POR
PROF. ME. VENILTON FALVO JR

ARARAQUARA 2019

JOAO CAMARGO

FROTCH

PIM III – PROJETO MULTIDISCIPLINAR III

Relatório final, apresentado a Universidade Paulista, como parte das exigências para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Araraquara, ____ de Junho de 2019.

BANCA EXAMINADORA

Prof. (Nome do orientador)

Afiliações

Prof. (Nome do professor avaliador)

Afiliações

Prof. (Nome do professor avaliador)

Afiliações

RESUMO

O PRESENTE TRABALHO TEM COMO OBJETIVO APRESENTAR
SOLUÇÕES PARA UMA EMPRESA DE SERVIÇOS DE FROTA DE
VEÍCULOS.

ABSTRACT

THE PRESENT WORK HAS TO OBJECTIVE TO PRESENT SOLUTIONS FOR
A COMPANY OF SERVICES OF FLEETING OF VEHICLES.

Sumário

1.	Introdução.....	1
1.1	Contextualização e Motivação.....	1
1.2	Objetivos	2
2.	Revisão Bibliográfica	3
2.1	Considerações iniciais	3
2.2	Benchmarking	3
2.2.1	Tipos de <i>Benchmarking</i>	3
2.3	Persona	4

2.4 BPMN (<i>Business Process Model and Notation</i>)	7
2.5 UML (<i>Unified Modeling Language</i>).....	8
2.6 RAD (<i>Rapid Application Development</i>)	15
2.7 Sustentabilidade.....	16
3. Requisitos do Sistema	17
3.1 Cenário	17
3.2 Processo de Negócio	17
3.3 Requisitos Funcionais e Não Funcionais	19
3.4 Caso de Uso	23
3.5 Descrições dos Casos de Uso	24
4. Projeto.....	49
4.1 Escolha da Metodologia	49
4.2 Diagramas de Classe.....	50
4.3 Diagramas de Sequência	53
4.4 Banco de Dados e DER	54
5. Prototipação.....	56
5.1 Benchmark	56
5.2 Persona	61
6. Conclusão	63
7. Bibliografia:	64

1. Introdução

1.1 Contextualização e Motivação

Atualmente o software tornou-se um elemento muito importante para o mundo e têm a capacidade de manipular as informações através de elementos computacionais e tiveram significativas mudanças ao longo do tempo e até hoje continuam tendo. Com o crescimento desse segmento muitas empresas possuem mais especialistas em TI em que cada um tem sua responsabilidade no desenvolvimento de software e é diferente de antigamente que era um único profissional de software que trabalhava sozinho em sala (PRESSMAN, 2016).

O software é o conjunto de vários artefatos e não apenas o código fonte (SOMMERVILLE, 2011) assume um duplo papel, ou seja, ele é um produto e, ao mesmo tempo, o veículo para entrega do produto. Esse produz, gera, adquire, modifica, exibe ou transmite informação. Isto é, o software funciona como um veículo de entrega do produto mais importante da época – a informação. Assim o software disponibiliza o potencial que a tecnologia traz tanto para ser utilizado em uma rede de computadores local, como para um canal amplo, como a internet. Segundo Pressman (2006), as aplicações Web cobrem uma ampla gama de domínios. Podem ser um conjunto simples de arquivos ligados por hipertexto e poucos gráficos. Por outro lado, em sua evolução, podem fornecer características de grandes softwares, como, funções de computação avançada e conteúdo para o usuário final, que podem ser integradas ao banco de dados da organização e se adequarem às aplicações do negócio. Essas organizações buscam a qualidade, eficácia e querem ter resultados positivos, para isso precisam se automatizar e controlar as suas rotinas administrativas.

Baseado no contexto e nas informações citadas anteriormente, a empresa de Telecomunicações Águias Brasil, decidiu automatizar a empresa para facilitar o dia-a-dia de seus funcionários e para melhor atender seus clientes, para isso foi contratada a empresa FROTCH Automação para desenvolver um sistema para gerenciar sua frota de veículos e controle de seus serviços gastos com manutenções nos mesmos.

1.2 Objetivos

O objetivo deste trabalho é o desenvolvimento de um sistema voltado ao ambiente Web para controle e gestão de uma frota de veículos. Tendo por objetivo geral desenvolver um Sistema para o cadastro e controle dos veículos e o gerenciamento de seus serviços e manutenções (preventiva ou eventual), e suas rotas (destino e quilometragens percorridas). O Sistema irá exercer as seguintes funcionalidades:

- Permitir o controle dos cadastros gerais de uma frota de veículos;
- Emitir relatórios analíticos para acompanhamento de toda a frota de veículos;
- Apresentar as reservas da frota de veículos para o dia, bem como as suas manutenções;
- Permitir gerar os relatórios diários de serviços efetuados e também os relatórios de cadastros gerais do sistema com os perfis de usuários, de veículos, de departamentos e de cidades;
- Permitir o controle dos cadastros de peças (entrada e saídas de itens para os serviços realizados).

Sendo assim, o sistema deverá utilizar o software em combinação com a tecnologia Web, provendo uma solução gerencial das movimentações em que envolvam o fluxo da frota de veículos, através de uma Web site.

Portanto, apresenta-se neste trabalho um sistema de informação que automatiza as operações efetuadas na empresa de Telecomunicações Águia e possibilita um controle eficiente sobre a frota de veículos. Este controle é obtido pelo acesso rápido às informações, podendo ser consultadas em telas do sistema ou visualizadas através da emissão de relatórios.

2. Revisão Bibliográfica

2.1 Considerações iniciais

Para o desenvolvimento deste projeto foram feitas pesquisas que auxiliaram na construção do projeto com mais ênfase. Nesse projeto foram utilizadas BPMN (*Business Process Model and Notation*) e UML (*Unified Modeling Language*) e desenvolvido com a metodologia RAD (*Rapid Application Development*), foram criadas Personas em técnicas de reconhecimento de usuários, e efetuado o processo de *Benchmark* para melhor entendimento do produto proposto. Sendo assim, segue um resumo das pesquisas as quais foram essenciais para elaboração e desenvolvimento usando um cronograma de acordo com o modelo de modelagem apresentado neste documento.

2.2 Benchmarking

A técnica de *Benchmarking* é um processo de comparação de produtos, serviços e práticas empresariais, e é um importante instrumento de gestão das empresas. Ele é realizado através de pesquisas que compararam as ações de uma determinada empresa e tem o objetivo de melhorar as funções e processos da empresa, além de ser um importante aliado para vencer a concorrência, uma vez que o *Benchmarking* analisa as estratégias e possibilita a outra empresa criar e ter idéias novas em cima do que já é realizado (SOUSA, 2018).

Para obtenção de um bom resultado no desenvolvimento deste projeto, esta técnica foi utilizada na construção deste trabalho e será demonstrada no Capítulo 5.

2.2.1 Tipos de *Benchmarking*

O Benchmarking consiste em aprender com outras empresas, sendo um trabalho de grande intensidade, que requer bastante tempo e disciplina. Pode ser aplicado a qualquer processo e é relevante para qualquer organização, tendo em

conta que se trata de um instrumento que vai contribuir para melhor o desempenho da empresa ou organização (SOUSA, 2018). É possível identificar 4 tipos de Benchmarking:

- Benchmarking competitivo;
- Benchmarking genérico;
- Benchmarking funcional;
- Benchmarking interno;

Benchmarking competitivo está relacionado com os processos e gestão de empresas concorrentes. É uma área delicada, pois empresas concorrentes defendem e escondem as práticas que as levam ao sucesso.

Benchmarking genérico consiste na comparação de parâmetros da funcionalidade das empresas, em aspectos como o tempo que um determinado produto demora a chegar ao cliente, desde que foi requisitado.

Benchmarking funcional é muitas vezes comparado ao genérico porque é relativo a um processo de atuação da empresa, como a distribuição.

Benchmarking interno consiste em tomar como referência as práticas e processos de outros setores dentro da própria empresa, e tentar apropriá-los ou melhorá-los para outros setores. No âmbito interno, o Benchmarking favorece a própria empresa, uma vez que não precisa ter custos com pesquisas externas, e é um processo mais fácil de ser executado (SOUSA, 2018).

2.3 Persona

Persona, ou *buyer persona*, é um perfil semi ficcional que representa o cliente ideal de uma empresa, são criados para ajudar a compreender melhor quem é o cliente e do que ele precisa. Esses personagens fictícios são criados para representar os diferentes tipos de usuário dentro de um alvo demográfico, atitude e/ou comportamento definido que poderia utilizar um site, uma marca ou produto de um modo similar. O termo Persona é usado amplamente em aplicações online e tecnológicas, bem como em publicidade, onde outros termos como retratos de persona também podem ser usados. (SANTOS, 2019). Neste documento no

Capítulo 5, seção 5.2, será abordado o tema, e serão demonstradas com ilustrações como as Personas para esse projeto foram desenvolvidas.

A criação da Persona é muito mais específica e demanda muito mais tempo do que a definição do público-alvo. É como se o público-alvo fosse o esboço e a Persona o trabalho finalizado.

Para as empresas o conhecido Público-alvo no qual abrange um grande número de pessoas que podem ser atingidas pelo produto ou serviço que a empresa oferece. Ele envolve questões demográficas, socioeconômicas e comportamentais que definem as ações do seu negócio.

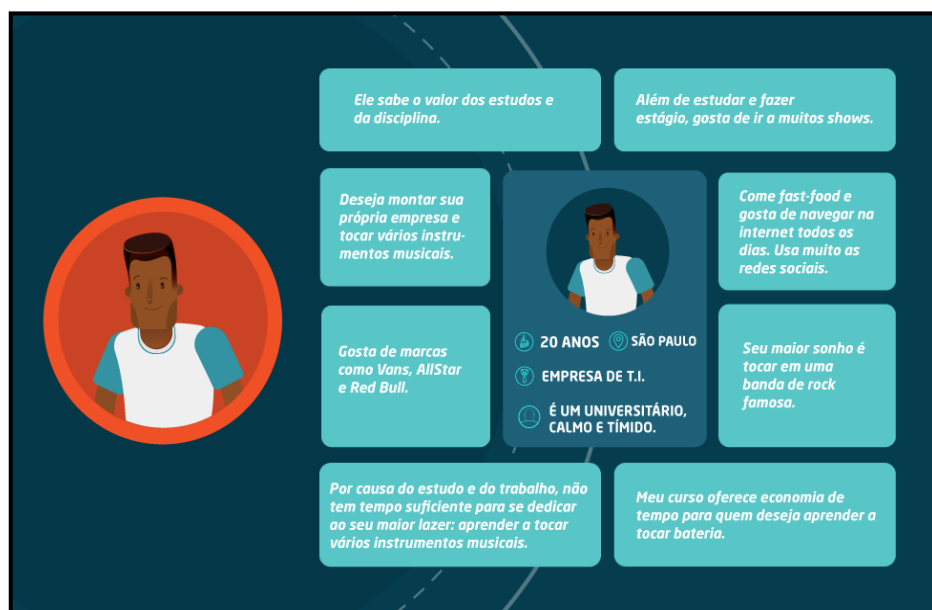
Qual a diferença entre Persona e Público-alvo?

Persona: Maurício, 35 anos, é gerente de marketing de uma grande empresa nacional. É solteiro, com uma renda mensal de R\$ 25.000. Mora em Curitiba e não tem muito tempo para navegar na internet por lazer.

Público-alvo: gerentes de marketing entre 30 e 45 anos que moram em capitais do Brasil. Trabalham em grandes empresas e não gastam muito tempo na internet para lazer. A renda mensal deles está entre R\$ 20.000 a R\$ 30.000. Percebe como a definição de público-alvo é muito mais ampla e não fala sobre hábitos e características de uma pessoa específica? Essa é a principal diferença entre Persona e público-alvo.

A Persona precisa ser tão real quanto uma pessoa de verdade, afinal, foi assim que foi criada. Um público-alvo não fornece informações, dados e insights suficientes para que uma boa estratégia de marketing digital seja criada.

Figura 1. Exemplo de como criar um script para gerar uma Persona.



Fonte:

<https://blog.hotmart.com/pt-br/como-criar-persona-negocio/>

Na Figura 1, foi criada uma lista das qualidades, gostos e costumes de um indivíduo para que assim fosse traçado um perfil para ser desenvolvido um Persona.

Figura 2. Exemplo de como criar uma Persona



Fonte: < <https://blog.hotmart.com/pt-br/como-criar-persona-negocio/> >

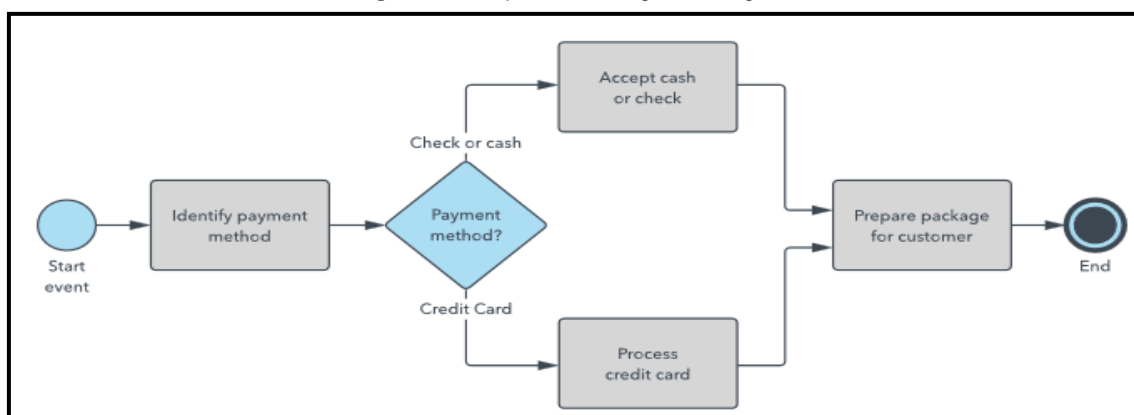
A Figura 2 representa um exemplo de Persona no qual foi criada baseada nos dados que foram extraídos das informações coletadas sobre um determinado

indivíduo, porém para dar veracidade aos fatos foi dada a esse personagem a identidade de Marcelo e demonstrados suas necessidades, e assim com a leitura sobre este tipo de perfil, pode ser desenvolvido um Software que atenda aos problemas enfrentados por ele.

2.4 BPMN (*Business Process Model and Notation*)

A Notação de Modelagem de Processos de Negócio (BPMN, em inglês) é um método de fluxograma que modela as etapas de ponta a ponta, de um processo de negócios planejado. Peça chave na gestão de processos de negócios representa de forma visual uma sequência detalhada de atividades de negócios e fluxos de informação necessários para concluir um processo (MAGALHÃES, 2019).

Figura 3. Exemplo de Modelagem de Negócio



Fonte: Lucidchart

Na Figura 3 demonstra um processo de pagamento no qual o usuário tem que efetuar o pagamento e para isso se criou um fluxo no qual há a possibilidade de serem feitos de 3 formas (dinheiro, cheque ou cartão de credito) e para isso o sistema gerou um fluxo alternativo com a pergunta qual método de pagamento e de acordo com a escolha do usuário o sistema segue para um caminho, efetua a ação e encerra o processo. O BPMN tem como finalidade modelar formas de melhorar a eficiência, representar novas circunstâncias ou ganhar vantagem competitiva. O método tem passado por uma nova padronização nos últimos anos, e agora é frequentemente chamado de: **Notação e Modelo de Processos de Negócio**, mas ainda mantém a sigla BPMN. Ele difere do mapeamento de processos de negócios, que, por sua vez, diagrama processos atuais para fins de padronização, treinamento de funcionários, gestão da qualidade e

conformidade de auditoria. BPMN também é uma contrapartida de negócios da UML, utilizada no design de software. **Fonte texto: Lucidchart**

O Capítulo 3.2 deste documento será tratado os processos de negócio da empresa contratante, pois através dessas informações serão analisadas quais são as partes que poderão ser automatizadas, sendo assim será a base

2.5 UML (*Unified Modeling Language*)

A finalidade da modelagem é apresentar uma visão lógica do sistema com suas funcionalidades, além de apresentar as interações que ocorrem no sistema, facilitando a compreensão e entendimento do mesmo, objetivando uma posterior implementação do sistema.

Segundo Booch et al. (2005),

“As linguagens de modelagem orientadas a objetos surgiram entre a metade da década de 1970 e o final da década de 1980, à medida que o pessoal envolvido com metodologia, diante de um novo gênero de linguagens de programação orientadas a objetos e de aplicações cada vez mais complexas, começou a experimentar métodos alternativos de análise e projeto. Aprendendo com a experiência, novas gerações desses métodos começaram a surgir, com um claro destaque de alguns como Booch, o OOSE (Object-Oriented Software Engineering) de Jacobson, e o OMT (Object Modeling Technique) de Rumbaugh. Entre outros métodos importantes encontravam-se Fusion, Shlaer-Mellor e Coad-Yourdon. Em termos simples, o método Booch era expressivo principalmente durante as fases de projeto e construção de sistema, o OOSE fornecia excelente suporte para os casos como uma maneira de controlar a captura de requisitos, a análise e o projeto em alto nível; o OMT-2 era mais útil para análise e sistema de informações com uso intensivo de dado”.

Na metade da década de 1990 foi surgindo uma massa crítica de idéias que vinha tomando forças, fazendo com que os principais métodos, Booch, OOSE e OMT dos quais já havia um reconhecimento mundial viessem a unir-se.

Em Outubro de 1994 foi criada a versão 0.8 da UML criada a princípio da junção de BOOCH e OMT. Após um ano, em Outubro de 1995, Jacobson se associa aos dois e expande o projeto para incorporar o OOSE, lançando assim em junho de 1996 a versão 0.9 da UML.

Em 1997 a UML foi apresentada ao OMG (*Object Management Group*), uma associação sem fins lucrativos que se dedica a manter as especificações a serem usadas pelas indústrias de computadores. E a partir dessa fusão seguiram versões cada vez mais melhoradas e adaptando-se aos tempos (BOOCH et al, 2005).

A UML tornou-se uma linguagem padrão utilizadas para modelagem de projetos e sistemas, que demonstra através de palavras e figuras, as funcionalidades do sistema proposto com visualização, especificação, construção e documentação dos elementos de um software facilitando assim o entendimento ao analista, programador ou até mesmo ao usuário.

Atualmente a UML é composta por vários diagramas (cerca de 13 diagramas) para representar as funcionalidades do sistema, porém neste Projeto serão usados os seguintes diagramas:

- Diagrama de Classe (*Class Diagram*);
- Diagrama de Casos de Usos e Descrição dos Casos de Uso (*Use Case Diagram*);
- Diagrama de Sequência (*Sequence Diagram*).

Diagrama de Classe: são utilizadas para modelar as classes, incluindo seus atributos, operações e associações com outras classes. Ele fornece uma visão estática ou estrutural de um sistema, mas não mostra a natureza dinâmica das comunicações entre os objetos das classes no diagrama (PRESSMAN, 2016).

As classes em um diagrama de classes podem representar os principais objetos, interações no aplicativo ou classes a serem programadas.

As Classes são templates para a criação de objetos e implementação de comportamento em um sistema. Em UML, uma classe representa um objeto ou um conjunto de objetos que compartilham uma estrutura e comportamento comum. São representadas por um retângulo que inclui linhas do nome da classe, seus atributos e suas operações. Ao desenhar uma classe em um diagrama de classes, somente a primeira linha deve ser preenchida e as outras são opcionais caso queira fornecer mais detalhes.

A representação gráfica das classes no Diagrama de Classe segue os seguintes passos:

Nome: a primeira linha em uma forma de classe.

Atributos: ficam na segunda linha que representa o corpo da Classe. Cada atributo da classe é exibido em uma linha separada e tem sua representatividade de acordo com sua visibilidade.

- Público (+)
- Privado (-)
- Protegido (#)
- Pacote (~)
- Derivado (/)
- Estático (sublinhado)

Métodos: também conhecidos como operações, os métodos são exibidos em formato de lista, com cada operação representada em sua própria linha.

Sinais: símbolos que representam comunicações unidirecionais e assíncronas entre objetos ativos.

Tipos de dados: classificadores que definem valores de dados. Os tipos de dados podem modelar tipos primitivos e também enumerações.

Pacotes: formas projetadas para organizar classificadores relacionados em um diagrama. São simbolizados por uma grande forma de retângulo com abas.

Interfaces: uma coleção de assinaturas de operações e/ou definições de atributos que definem um conjunto coeso de comportamentos. Interfaces são semelhantes às classes, exceto que uma classe pode ter uma instância de seu tipo, e uma interface deve ter pelo menos uma classe para implementá-la.

Enumerações: representações de tipos de dados definidos pelo usuário. Uma enumeração inclui grupos de identificadores que representam os valores da enumeração.

Objetos: instâncias de uma classe ou classes. Objetos podem ser adicionados a um diagrama de classes para representar instâncias concretas ou prototípicas.

Artefatos: elementos de modelo que representam as entidades concretas em um sistema de software, tais como documentos, bancos de dados, arquivos executáveis, componentes de software etc.

Herança: Um conceito muito importante em design orientado a objetos, herança, refere-se à capacidade de uma classe (classe-filha) de herdar a funcionalidade idêntica de outra classe (superclasse) e, em seguida, incluir sua nova funcionalidade própria. Para modelar a herança em um diagrama de classes, uma linha sólida é desenhada a partir da classe-filha (a classe que herda o comportamento) com uma ponta de seta (ou um triângulo) fechada, não preenchida, apontando para a superclasse.

Implementação: (utilizado apenas na perspectiva de implementação). Utilizado para indicar que uma classe implementa uma interface.

Para representar as conexões entre as classes foram criados alguns elementos que ajudam no entendimento do Diagrama, segue abaixo uma lista dos elementos gráficos utilizados para a representação de Associação, Herança, Dependência, Agregação e de Composição.

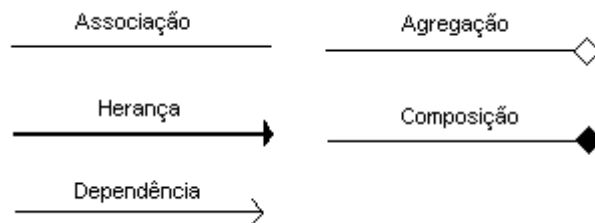
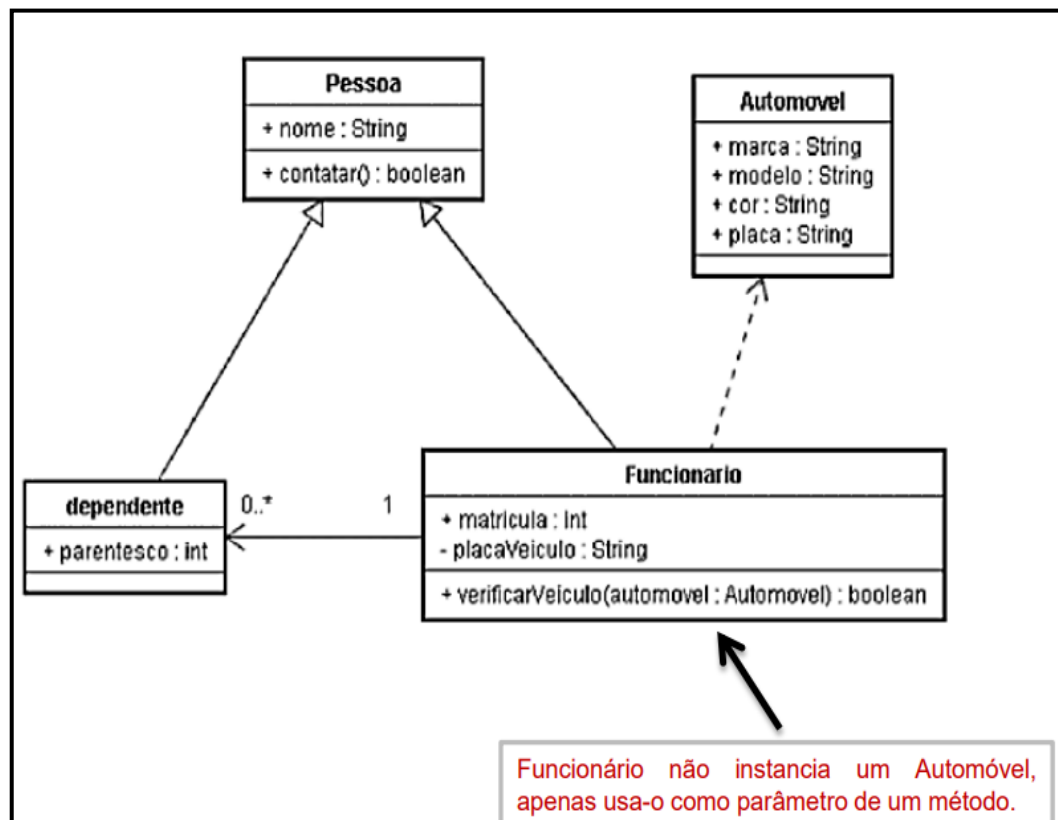


Figura 4. Diagrama de Classe com interação entre as classes.



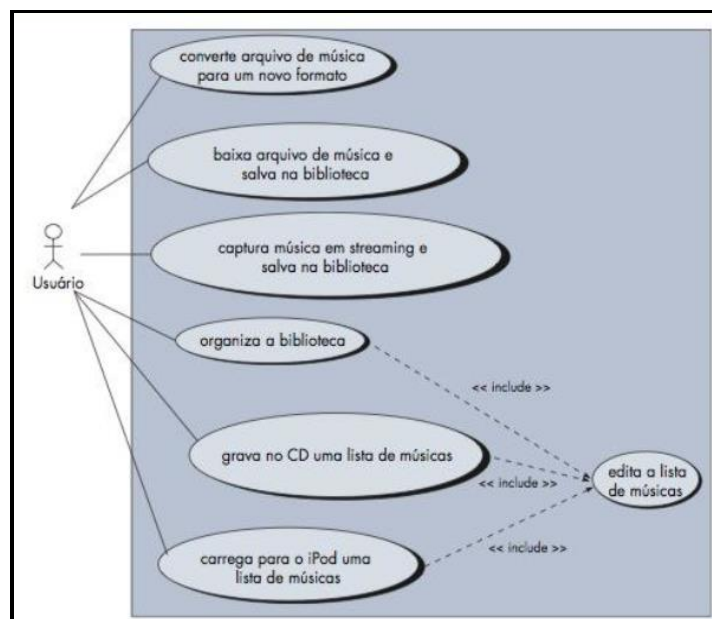
Fonte: Apêndice 1 – Introdução a UML – Pressman (2016).

Na Figura 4 estão sendo demonstrados os conceitos de herança onde as Classes dependentes e *Funcionario* estão herdando os atributos da Classe Pessoa. Representa também que a Classe *Automovel* é uma implementação de interface da Classe *Funcionario*. Sendo assim as ligações de relacionamentos entre as classes são representadas pelas setas conforme demonstra a Figura 4.

Diagrama de Caso de Uso: Casos de Uso e o Diagrama de Caso de Uso servem para determinar a funcionalidade e as características do software sob o ponto de vista do usuário. Um Caso de Uso descreve como o usuário irá interagir com o sistema mostrando cada passo necessário para chegar ao seu objetivo. O diagrama de Caso de Uso é uma visão geral de todos os Casos de Uso

mostrando a funcionalidade de cada Ator (Usuário) dentro do Sistema. (PRESSMAN, 2016).

Figura 5. Diagrama de Caso de Uso para sistema de musica



Fonte: Apêndice 1 – Introdução a UML – Pressman (2016).

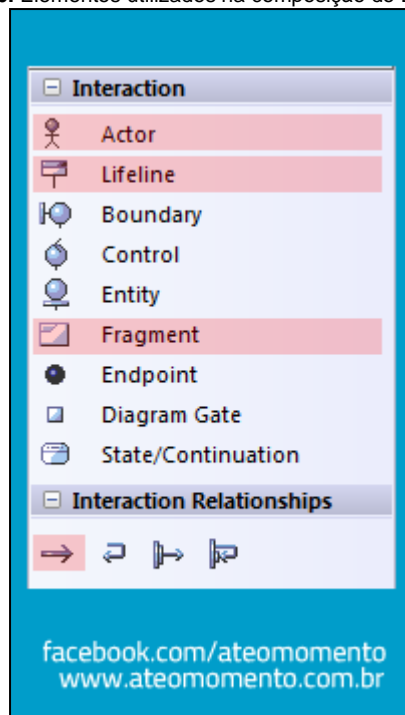
Na Figura 5 representa um diagrama de Caso de Uso onde um Ator (Usuário) desempenha várias funcionalidades. Essas funcionalidades são representadas pelos “balões” onde são inseridas as Ações que serão exercidas pelo Ator. As ligações de Associação são representadas por uma linha contínua. A relação que representa uma inclusão de ação da desempenhada inicialmente está representada pela linha tracejada com uma ponta de seta.

Neste documento no capítulo 3.4 e 3.5 serão demonstrados os Casos de Uso e suas Descrições, respectivamente.

Diagrama de Sequência: este diagrama serve para mostrar as comunicações dinâmicas entre os objetos durante a execução de uma tarefa. Ele mostra a ordem e o tempo no qual as mensagens são enviadas entre os objetos para executar aquela tarefa (PRESSMAN, 2016).

Na UML temos três conceitos necessários de entender: diagramas, elementos e relacionamentos. As formas gráficas que compõe cada diagrama são chamadas “elementos”. Cada elemento possui um objetivo específico, e a combinação destes elementos torna-se o diagrama, que gera a semântica do respectivo modelo.

Figura 6. Elementos utilizados na composição do Diagrama



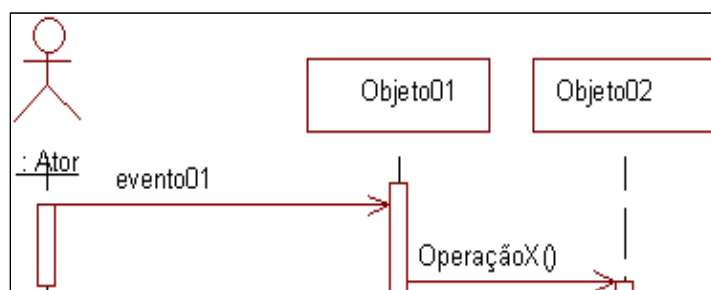
Actor (Ator) – É o usuário que inicia a interação, a troca de mensagens. Pode ser um ator humano (usuário de sistema), uma funcionalidade ou componente (usuário sistêmico, o sistema chamando algum componente).

Lifeline (Linha de Vida) – É a instância de um componente (veremos mais nos exemplos a seguir), aonde chegam chamadas, e de onde partem chamadas (chamadas = mensagens).

Fragment (Fragmento) – É uma “caixa” que inserimos nos diagramas de sequência onde destacamos estruturas condicionais (if/else), loops (for/while), tratamentos de exceção etc.

Message (Mensagem) – É a mensagem “de fato”. Ligamos a “seta” nas lifelines (linhas de vida/instâncias dos componentes) contidas no diagrama (sempre na direção do fluxo das interações), definindo qual é o método usado (se um diagrama de projeto), parâmetros/valores passados nas mensagens. Se é um diagrama de esboço (para troca de ideias) então as mensagens são mais textuais.

Figura 7. Diagrama de Sequencia



Fonte: Elaborada pelo autor.

Na Figura 7 está sendo representada uma interação entre o Ator e o Objeto01, onde uma ação (evento) tem seu início e é representado pela seta continua em sentido ao Objeto01. O Diagrama também demonstra uma interação entre o Objeto01 e o Objeto02. Em relação a seta que retorna a mesma caixa de ativação é um auto relacionamento no qual um método traz um retorno ao Ator.

2.6 RAD (*Rapid Application Development*)

Para o desenvolvimento do projeto foi adotado a metodologia *Rapid Application Development* (RAD), no qual inúmeras empresas do ramo de desenvolvimento utilizam devido à praticidade e agilidade no qual o mesmo proporciona. Este método de modelagem geralmente é utilizado para projetos com curtos prazos tempo para a sua entrega. Segue abaixo uma breve explicação sobre o assunto abordado:

“Rapid Application Development (RAD) é um chavão de marketing que quase todas as ferramentas de desenvolvimento de software usam, mas que raramente se aplica. Em um nível alto, é uma técnica de desenvolvimento de aplicativos que usa Protótipos iterativos, Personalização e Ferramentas Case. RAD é uma metodologia de desenvolvimento de software que se concentra na construção de aplicações em um

período muito curto de tempo, tradicionalmente, com compromissos em usabilidade, recursos e/ou velocidade de execução. (BLUEINK, 2016).

No desenvolvimento do projeto também foram utilizadas ferramentas de programação Eclipse e o Banco de Dados Oracle, qual foram de essencial importância no desenvolvimento (programação) do projeto.

2.7 Sustentabilidade

Adotar práticas sustentáveis na gestão de frotas, além de reduzir consumo e custo, aumenta o valor da empresa, que demonstra ter consciência e responsabilidade ambiental. Fontes de energia renováveis são literalmente encontradas na luz do sol, no ar, no subsolo profundo e nos oceanos. Eles são parte da estrutura física do planeta, o que significa que eles estão sendo constantemente renovados por meios naturais, ou seja, simplesmente não podem acabar. Essas fontes de energia sustentáveis são frequentemente chamadas de “energia alternativa” porque são consideradas uma alternativa aos combustíveis fósseis tradicionais, como o petróleo e o carvão. A bioenergia é um tipo de energia renovável derivada da biomassa para criar calor e eletricidade ou para produzir combustíveis líquidos, como o etanol e o biodiesel, usados no transporte.

Embora a bioenergia gere aproximadamente a mesma quantidade de dióxido de carbono que os combustíveis fósseis, as plantas de substituição cultivadas como biomassa removem uma quantidade igual de CO₂ da atmosfera, mantendo o impacto ambiental relativamente neutro. Há uma variedade de sistemas usados para gerar esse tipo de eletricidade, desde a queima direta da biomassa até a captura e uso do gás metano produzido pela decomposição natural do material orgânico. No Brasil os investimentos totais previstos na área de biocombustíveis somam R\$ 97 bilhões até o final desta década, concentrados quase que integralmente na oferta de etanol.

Adotar práticas de sustentabilidade reduz os custos para a empresa devida:

- Redução na emissão de gases, diminuindo os impactos ambientais;
- Maior faturamento, se considerarmos que empresas verdes conquistam mais consumidores e podem ter vantagens em processos licitatórios;
- Redução de custos com manutenção;
- Aumento da vida útil do veículo;
- Maior segurança aos motoristas, que tendem a dirigir de forma mais consciente e dentro da velocidade permitida, por exemplo;
- Maior produtividade, considerando que há um planejamento de roteiros e utilização das frotas;
- Apoio para tomada de decisão, já que o histórico das informações fica registrado e pode ser analisado;
- Maior controle das multas recebidas, evitando o pagamento de multas desnecessárias.

Sendo assim o sistema proposto irá auxiliar a empresa contratante a executar essas ações, a fim de reduzir seus gastos e contribuir para um ambiente mais saudável.

3. Requisitos do Sistema

3.1 Cenário

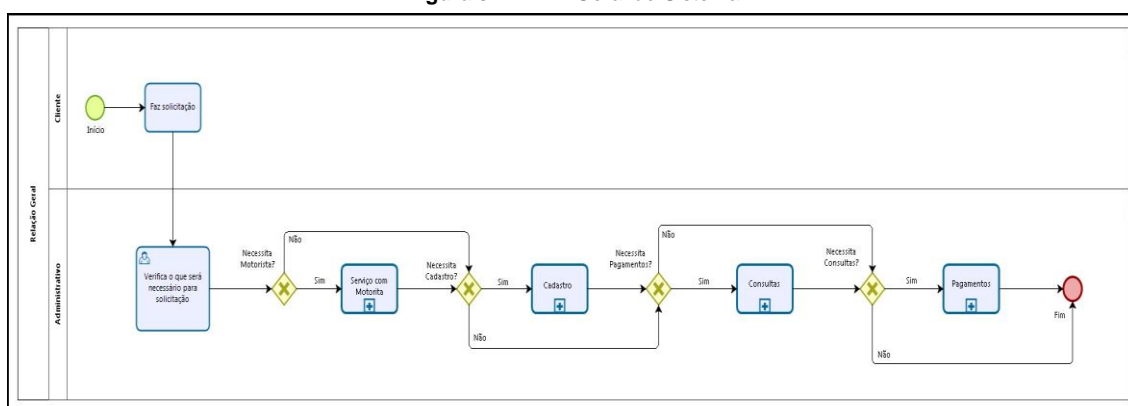
Segundo Pressman (2016), a especificação do software é o produto do trabalho final, ele descreve a função e o desempenho das atividades do software. Para esta demonstração das especificações do sistema utilizou-se a notação do BPMN no qual foi diagramado o processo de negócio da empresa e também foram utilizadas UML, sendo os diagramas de Caso de Uso no qual foram gerados através da ferramenta de modelagem online *Creately*⁽¹⁾, e suas descrições para elucidação dos caso de uso identificados.

3.2 Processo de Negócio

Para ilustrar o processo de negócio da empresa solicitante, foi necessária a criação de uma modelagem de processo de negócio, também conhecido como BPMN e, portanto, foi necessária uma pesquisa detalhada das atividades da empresa em questão e subdividir as atividades em 4 módulos, no qual serão detalhados a seguir neste capítulo.

Para a demonstração geral do processo, foi criado um BPMN usando a ferramenta *Bizagi*, onde está sendo mostrado como é efetuada a solicitação de um serviço na empresa e quais são os processos pelos quais esse evento tem que passar até a atividade ser encerrada.

Figura 8. BPMN Geral do Sistema

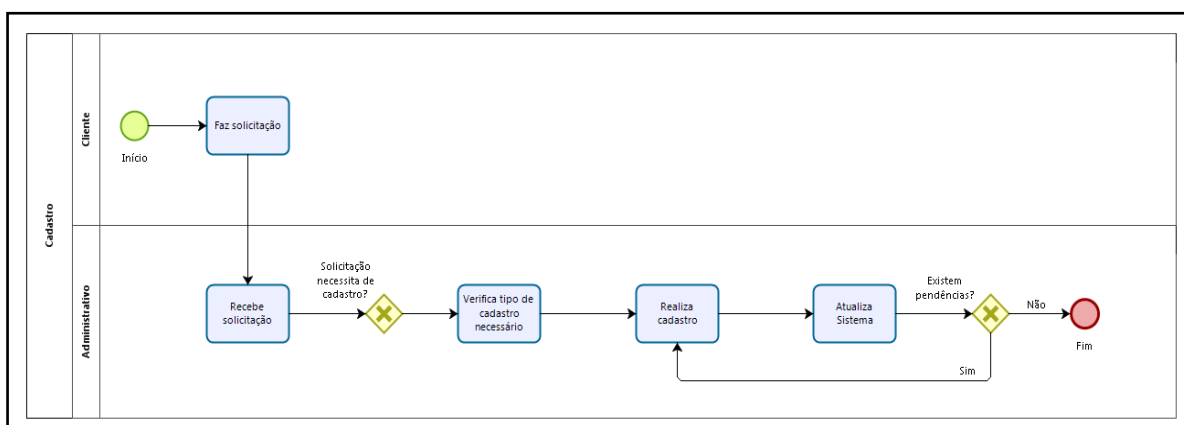


Fonte: Elaborada pelo autor. Disponível em: <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

¹ Disponível em: <https://creately.com>

Conforme está demonstrado na Figura 8, o cliente efetua um pedido a empresa de telefonia e o atendente da empresa por sua vez irá analisar quais serão os procedimentos a serem realizados, se for necessário irá deslocar um veículo até o local onde está ocorrendo o problema citado pelo cliente, porém para isso será necessário ter um motorista disponível para esse serviço, se tiver algum motorista disponível, o atendente irá efetuar o cadastro do cliente e solicitar qual será a forma de pagamento (à vista ou parcelado) logo após o atendente irá confirmar o pedido, finalizando a operação.

Figura 9. BPMN de Cadastro



Fonte: Elaborada pelo autor. **Disponível em:** <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

Na Figura 9 esta diagramada a sequência de cadastro de um cliente no sistema, no qual será feita a solicitação do cadastro pelo cliente ao atendente e este por sua vez irá inserir os dados no sistema. Após serem inseridos os dados, o sistema fará as verificações necessárias e retorna se será necessário o cadastro ou alteração de um cadastro já.

Para o desenvolvimento do sistema foram necessários outros diagramas de BPMN, no qual estão disponíveis em arquivo da plataforma no repositório em nuvem no site do *OneDrive*⁽¹⁾. O repositório foi criado devido a necessidade de arquivar conteúdos que não couberam na impressão devido a sua extensão, e para não perder a qualidade nas imagens armazenadas nele.

3.3 Requisitos Funcionais e Não Funcionais

De acordo com Pressman (2016), os requisitos são na verdade uma ponte entre o projeto e a construção do sistema, é um processo que identifica as necessidades do negócio e as restrições do projeto. Ou seja, com os requisitos é possível que o desenvolvimento do sistema tenha um ponto de partida. O Quadro 1 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com os casos de uso associados.

Quadro 1: Requisitos funcionais

Requisitos Funcionais	Caso de Uso
-----------------------	-------------

RF01	<p>O sistema deverá permitir ao administrador incluir, alterar e excluir colaboradores. Cadastrar com os dados: (Nome, endereço, telefone, data de nascimento, CTPS, CPF, RG, Data do registro, Cargo, Salário);</p> <p>1.1 - Caso o cargo do funcionário for motorista o sistema deve-se solicitar que insira os dados da CNH e data de seu vencimento, exames médicos;</p>	UC02
RF02	<p>Para ter acesso ao sistema, o colaborador deve informar login e senha de identificação.</p> <p>2.1 - O sistema deverá possuir dois perfis de usuário. Cada um com diferentes níveis de privilégio.</p>	UC19
RF03	O sistema deverá permitir ao administrador incluir, alterar e excluir relatórios;	UC17
RF04	O sistema deverá permitir ao administrador incluir, alterar e excluir veículos. Cadastrar os veículos com os seguintes dados: (Marca e Modelo, cor, placa);	UC01
RF05	O sistema deverá permitir ao administrador incluir, alterar e excluir categorias de veículos (Próprio ou Alugado);	UC01
RF06	O sistema deverá permitir ao administrador incluir, alterar e excluir marcas de veículos;	UC01
RF07	O sistema deverá permitir ao administrador incluir, alterar e excluir cidades;	UC11
RF08	O sistema deverá permitir ao colaborador consultar, incluir, alterar, finalizar e cancelar serviços;	UC04
RF09	O sistema deverá permitir ao administrador inserir, alterar e excluir manutenções de veículos;	UC09
RF10	O sistema deverá permitir ao administrador a consulta de colaboradores ativos e inativos;	UC02
RF11	O sistema deve permitir o administrador incluir, alterar, consultar e excluir valores de ocorrências de sinistros ocorridos armazenando os dados:	UC09

	(condutor, veículo, valor, data de pagamento do seguro, se teve vítimas);	
RF12	O sistema deverá permitir ao administrador a consulta de veículos ativos e inativos;	UC10
RF13	O sistema deverá permitir ao administrador a consulta de cidades ativas e inativas;	UC13
RF14	O sistema deverá permitir ao administrador emitir relatório de serviços filtrando por período, veículo e totalizando por quilômetros rodados;	UC17
RF15	O sistema deverá permitir ao administrador o cadastramento de peças adquiridas;	UC03
RF16	O sistema deverá permitir ao administrador emitir relatório de manutenções.	UC17
RF17	O sistema deverá permitir ao administrador emitir relatórios estatísticos de veículos e colaboradores;	UC17
RF18	O sistema deverá permitir ao colaborador efetuar o <i>login</i> para acessar as funcionalidades do sistema;	UC19
RF19	O sistema deverá permitir ao colaborador consultar, incluir, alterar, finalizar e excluir serviços;	UC04
RF20	O sistema deverá permitir ao colaborador visualizar os veículos disponíveis por data e horário para deslocamento de serviços;	UC10
RF21	O sistema deverá permitir ao colaborador consultar, incluir, alterar e excluir alertas de manutenção de veículos;	UC04
RF22	O sistema deve permitir que o administrador possa inserir, alterar, consultar, excluir dados dos Seguros dos veículos;	UC09
RF23	O sistema deve armazenar dados de viagens efetuadas com o veículo (rotas, ocupantes do veículo, data e horário, quilometragens);	UC11
RF24	O sistema deve gerar relatórios que auxiliem controlar as finanças (compra de peças e	UC17

	pagamentos de serviços feitos a oficinas credenciadas);	
RF25	O sistema deve gerar relatórios por período para controlar valores gastos com abastecimentos por veículos;	UC17
RF26	O sistema deve permitir o administrador incluir, alterar, consultar e excluir valores de ocorrências de multas com dos dados: (condutor, veículo, valor, data de pagamento);	UC09

O Quadro 2 apresenta os requisitos não funcionais previstos para o sistema.

Quadro 2: Requisitos não funcionais

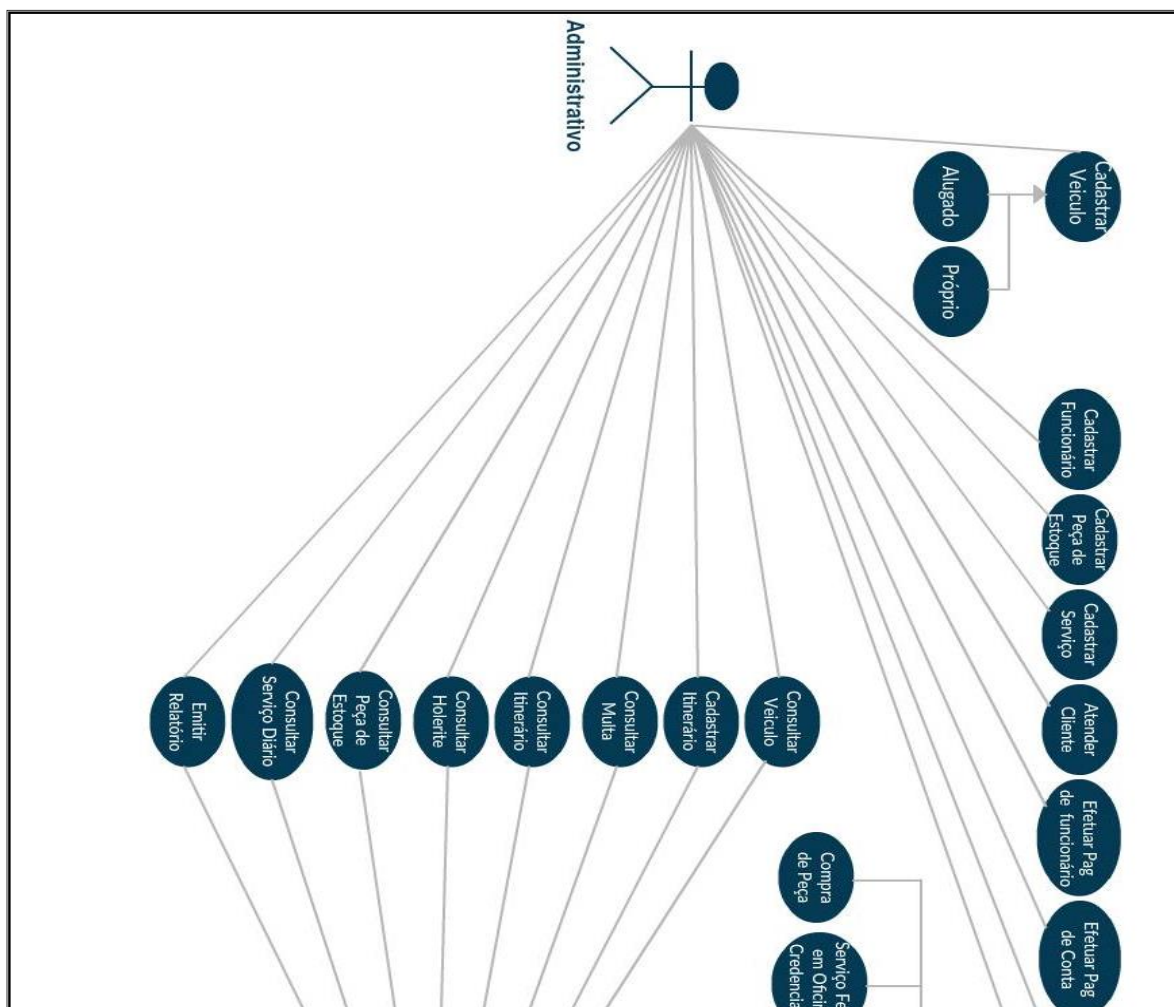
Requisitos não Funcionais		Caso de Uso
RNF01	O sistema deverá utilizar banco de dados desenvolvido em Oracle.	
RNF02	O sistema deverá ser desenvolvido utilizando a linguagem de programação HTML, CSS, Java Script, PHP.	
RNF03	Aplicação suportará os browsers Google Chrome, Mozilla Firefox 1.5, Opera e Internet Explorer 9 ou superior	
RNF04	O servidor de internet deve ter no mínimo processador Pentium 1GHz (ou equivalente) com o mínimo de 1 GB de memória (ou superior).	
RNF05	O sistema apresentará uma interface para situar o colaborador dentro do site.	

RNF06	O sistema deve ter uma criptografia confiável como Blowfish ou MD5	
RNF07	O sistema deve ter acessibilidade para pessoas com deficiência visual, tendo recursos de aumento das fontes e ter contraste na paleta de cores.	
RNF08	O sistema deve permitir a mudança de idioma, e por consequência tradução do sistema.	

3.4 Caso de Uso

Para o melhor entendimento do Sistema foi utilizada a modelagem UML com a ferramenta online *CreateUML* para descrever as funções de cada Ator dentro do Sistema. Através dos Casos de Uso foi descrito como o Sistema deverá agir em suas funcionalidades.

Figura 10. Diagrama de Casos de Uso



Fonte: Elaborada pelo autor. **Disponível em:** <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

Na figura 10 estão representadas as funções dos Atores Administradores e os Atores Motoristas e qual ação cada um deles poderão efetuar quando estiverem logados no Sistema.

O Diagrama de Caso de Uso na UML é um diagrama comportamental. Mas quando o assunto é caso de uso, o diagrama é só uma parte da solução. O principal mesmo nem é o diagrama (parte gráfica), mas sim a especificação do caso de uso (o que tem “dentro de cada bolinha”), a descrição dos seus cenários, no será tratado no próximo tópico deste documento.

3.5 Descrições dos Casos de Uso

Caso de Uso: UC01
Breve Descrição do Caso de Uso: Cadastrar Veículo.
Atores: Administrativo.
Pré-condição: Veículo não consta no banco de dados.
Fluxo Básico:

1. O administrativo inicia o caso de uso “Cadastrar Veículo”. [A1]
2. O sistema solicita marca/modelo, cor, placa, chassi, RENAVAM, categoria (Próprio ou alugado), situação (Garagem ou em serviço), combustível e motor.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema mostra uma mensagem “Veiculo cadastrado com sucesso”.
5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere os dados no sistema.
4. O sistema mostra uma mensagem “Veiculo cadastrado com sucesso”.
5. Encerra o caso de uso.

Caso de Uso: UC02
Breve Descrição do Caso de Uso: Cadastrar Funcionário.

Atores: Administrativo.

Pré-condição: Funcionário não consta no banco de dados.

Fluxo Básico:

1. O administrativo inicia o caso de uso “Cadastrar Funcionário”. [A1]
2. O sistema solicita Nome, endereço, telefone, data de nascimento, naturalidade, CTPS, CPF, RG, título de eleitor, e-mail, estado civil, grau de escolaridade, Cargo e Salário.
3. O administrativo pede ao funcionário os dados.
4. O administrativo insere os dados no sistema. [E1] [E2]
5. O sistema mostra uma mensagem “Funcionário cadastrado com sucesso”.
6. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro “Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere os dados no sistema.

4. O sistema mostra uma mensagem “Funcionário cadastrado com sucesso”.
5. Encerra o caso de uso.

[E2]: Cargo “Motorista”.

1. O sistema mostra uma mensagem que para o cargo “Motorista”, são necessários os dados como dados da CNH e data de seu vencimento e exames médicos.
1. O sistema mostra os campos necessários para inserir esses dados.
2. O administrativo insere esses dados.
3. O administrativo retoma o fluxo básico.

Caso de Uso: UC03

Breve Descrição do Caso de Uso: Cadastrar Peça de Estoque.

Atores: Administrativo.

Pré-condição: Peça não consta no banco de dados.

Fluxo Básico:

- 1.O administrativo inicia o caso de uso “Cadastrar peça”. [A1]
- 2.O sistema solicita o nome, modelo, código de barras, tipo, categoria, quantidade e data da entrada no estoque.
- 3.O administrativo insere os dados no sistema. [E1]
- 4.O sistema mostra uma mensagem “Peça cadastrada com sucesso”.
- 5.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere esses dados.
4. O sistema mostra uma mensagem “Peça cadastrada com sucesso”.
5. Encerra o caso de uso.

Caso de Uso: UC04

Breve Descrição do Caso de Uso: Cadastrar Serviço.

Atores: Administrativo.

Pré-condição: Serviço não consta no banco de dados.

Fluxo Básico:

1. O administrativo inicia o caso de uso “Cadastrar serviço”. [A1]

2. O sistema solicita nome do cliente, endereço, cidade, telefone, CPF/CNPJ, CEP, estado, e-mail, tipo do serviço, data para o serviço ser realizado e descrição do serviço.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema mostra uma mensagem "Serviço cadastrado com sucesso".
5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem "Deseja realmente cancelar?".
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro "Dados inválidos".
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um " * ".
3. O administrativo insere esses dados.
4. O sistema mostra uma mensagem "Serviço cadastrado com sucesso".
5. Encerra o caso de uso.

Caso de Uso: UC05

Breve Descrição do Caso de Uso: Atender Cliente.

Atores: Administrativo.

Pré-condição: Cliente solicita atendimento.

Fluxo Básico:

1. O administrativo inicia o caso de uso “Atender cliente”. [A1]
2. O sistema solicita nome do cliente, código, endereço, cidade, telefone, CPF/CNPJ, CEP, estado e e-mail.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema mostra uma mensagem “Cliente encontrado”.
5. O sistema mostra todas as informações referente ao cliente.
6. O administrativo realiza o atendimento.
7. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere esses dados.
4. O sistema mostra uma mensagem “Cliente encontrado”.
5. O sistema mostra todas as informações referente ao cliente.

6. O administrativo realiza o atendimento.
7. Encerra o caso de uso.

[E2]: Cliente não encontrado.

1. O sistema da uma mensagem "Cliente não encontrado".
2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC06

Breve Descrição do Caso de Uso: Efetuar Pagamento de Funcionário.

Atores: Administrativo.

Pré-condição: Ter o funcionário cadastrado no banco de dados.

Fluxo Básico:

1. O administrativo, inicia o caso de uso "Efetuar pagamento de funcionário". [A1]
2. O sistema solicita nome, CPF, RG, data de nascimento cargo do funcionário.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema retorna o funcionário especificado. [E2]
5. O administrativo clica no botão "Realizar depósito".
6. O Sistema mostra uma mensagem "Deposito realizado com sucesso".
7. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere esses dados.
4. O sistema retorna o funcionário especificado.
5. O administrativo clica no botão “Realizar deposito”.
6. O Sistema mostra uma mensagem” Deposito realizado com sucesso”.
7. Encerra o caso de uso.

[E2]: Funcionário não encontrado.

1. O sistema da uma mensagem de erro” Funcionário não encontrado”.
2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC07

Breve Descrição do Caso de Uso: Efetuar Pagamento de Conta.

Atores: Administrativo.

Pré-condição: Ter a conta cadastrada no banco de dados.

Fluxo Básico:

1. O administrativo, inicia o caso de uso “Efetuar pagamento de conta”. [A1]
2. O sistema solicita tipo de conta, data de emissão, data de vencimento, código de barras e valor.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema retorna a conta especificada. [E2]
5. O sistema informa se deseja pagar online ou fazer Download da conta.
6. O administrativo escolhe a opção desejada.
7. O administrativo realiza as operações de acordo com cada opção.
8. O sistema mostra na tela uma mensagem “Operação realizada com sucesso”.
9. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro "Dados inválidos".
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um " * ".
3. O administrativo insere esses dados.
4. O sistema retorna a conta especificada.
5. O sistema informa se deseja pagar online ou fazer Download da conta.
6. O administrativo escolhe a opção desejada.
7. O administrativo realiza as operações de acordo com cada opção.
8. O sistema mostra na tela uma mensagem "Operação realizada com sucesso".
9. Encerra o caso de uso.

[E2]: Conta não encontrada.

1. O sistema da uma mensagem de erro "Conta não encontrada".
2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC08

Breve Descrição do Caso de Uso: Lançar Holerite.

Atores: Administrativo.

Pré-condição: Ter o funcionário cadastrado no banco de dados.

Fluxo Básico:

1. O administrativo, inicia o caso de uso "Lançar holerite". [A1]
2. O sistema solicita nome, CTPS, cargo, valor de salário bruto, valor do salário líquido, horas extras e descontos de benefícios.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema mostra uma mensagem "Holerite cadastrado com sucesso".

5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro “Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere os dados no sistema.
4. O sistema mostra uma mensagem “Holerite cadastrado com sucesso”.
5. Encerra o caso de uso.

Caso de Uso: UC09

Breve Descrição do Caso de Uso: Cadastrar Conta.

Atores: Administrativo.

Pré-condição: Conta não consta no banco de dados.

Fluxo Básico:

1. O administrativo inicia o caso de uso “Cadastrar conta”. [A1]

2. O sistema solicita o tipo de conta (compra de peças, serviços feitos a oficinas credenciadas, abastecimentos, seguros, sinistro, multas e outros), solicitando o nome do funcionário, CPF, cargo, data de emissão, data de vencimento, código de barras, código da nota fiscal e valor.
3. O administrativo insere os dados no sistema. [E1]
4. O sistema mostra uma mensagem “Conta cadastrada com sucesso”.
5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo quer cancelar o caso de uso.
2. O administrativo clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar? ”.
4. O administrativo clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo insere esses dados.
4. O sistema mostra uma mensagem “Conta cadastrada com sucesso”.
5. Encerra o caso de uso.

Caso de Uso: UC10
Breve Descrição do Caso de Uso: Consultar veículo.

Atores: Administrativo e Motorista.

Pré-condição: Ter o veículo cadastrado no banco de dados.

Fluxo Básico:

1. O administrativo ou Motorista, inicia o caso de uso “Consultar veículo”. [A1]
2. O sistema solicita código do veículo, RENAVAM, placa e categoria (Próprio ou alugado).
3. O administrativo ou Motorista insere os dados no sistema. [E1]
4. O sistema retorna o veículo especificado.
5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo ou motorista quer cancelar o caso de uso.
2. O administrativo ou motorista clica no botão de cancelar.
3. O sistema mostra uma mensagem “Deseja realmente cancelar?”.
4. O administrativo ou motorista clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema da uma mensagem de erro” Dados inválidos”.
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
3. O administrativo ou Motorista insere os dados no sistema.
4. O sistema retorna o veículo especificado.
5. Encerra o caso de uso.

[E2]: Veículo não encontrado.

1. O sistema mostra uma mensagem” Veículo não encontrada”.

2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC11

Breve Descrição do Caso de Uso: Cadastrar Itinerário.

Atores: Administrativo e Motorista.

Pré-condição: Itinerário não consta no banco de dados.

Fluxo Básico:

- 1.O administrativo ou Motorista, inicia o caso de uso “Cadastrar itinerário”.
[A1]
- 2.O sistema solicita tipo de itinerário, Motorista, veículo, situação, gastos, Km inicial, Km final e descrição da viagem.
- 3.O administrativo ou Motorista insere os dados no sistema. [E1]
- 4.O sistema mostra uma mensagem “Itinerário cadastrado com sucesso”
- 5.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro” Dados inválidos”.
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
- 3.O administrativo ou Motorista insere os dados no sistema.
- 4.O sistema mostra uma mensagem “Itinerário cadastrado com sucesso”.
- 5.Encerra o caso de uso.

Caso de Uso: UC12

Breve Descrição do Caso de Uso: Consultar Multa.

Atores: Administrativo e Motorista.

Pré-condição: Ter a multa cadastrada no banco de dados.

Fluxo Básico:

- 1.O administrativo ou Motorista, inicia o caso de uso “Consultar multa”. [A1]
- 2.O sistema solicita código da infração, local da infração, município da infração e data da infração.
- 3.O administrativo ou Motorista insere os dados no sistema. [E1]
- 4.O sistema retorna a multa especificada. [E2]
- 5.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro" Dados inválidos".
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um " * ".
- 3.O administrativo ou Motorista insere os dados no sistema.
- 4.O sistema retorna a multa especificada. [E2]
- 5.Encerra o caso de uso.

[E2]: Multa não encontrada.

- 1.O sistema da uma mensagem de erro" Multa não encontrada".
- 2.O sistema pede para verificar os dados corretamente.
- 3.Retoma o passo 1 do fluxo básico.

Caso de Uso: UC13

Breve Descrição do Caso de Uso: Consultar Itinerário.

Atores: Administrativo e Motorista.

Pré-condição: Ter o itinerário cadastrado no banco de dados.

Fluxo Básico:

- 1.O administrativo ou Motorista, inicia o caso de uso "Consultar itinerário".
[A1]
- 2.O sistema solicita tipo de itinerário, Motorista, situação e veículo.
- 3.O administrativo ou Motorista insere os dados no sistema. [E1]
- 4.O sistema retorna o itinerário especificado. [E2]
- 5.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro” Dados inválidos”.
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
- 3.O administrativo ou Motorista insere os dados no sistema.
- 4.O sistema retorna o itinerário especificado.
- 5.Encerra o caso de uso.

[E2]: Itinerário não encontrado.

- 1.O sistema da uma mensagem de erro” Itinerário não encontrada”.
- 2.O sistema pede para verificar os dados corretamente.
- 3.Retoma o passo 1 do fluxo básico.

Caso de Uso: UC14**Breve Descrição do Caso de Uso:** Consultar Holerite.**Atores:** Administrativo e Motorista.**Pré-condição:** Ter o funcionário cadastrado no banco de dados.**Fluxo Básico:**

- 1.O administrativo ou motorista inicia o caso de uso “Consultar holerite”.
[A1]
- 2.O sistema solicita o código do funcionário, nome, CPF e cargo.
- 3.O administrativo ou motorista insere os dados no sistema. [E1]
- 4.O administrativo ou motorista clica no botão ok.
- 5.O sistema retorna o holerite especificado. [E2]
- 6.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro” Dados inválidos”.
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
- 3.O administrativo ou motorista insere os dados no sistema.
- 4.O administrativo ou motorista clica no botão ok.
- 5.O sistema retorna o holerite especificado.

6. Encerra o caso de uso.

[E2]: Holerite não encontrado.

1. O sistema dá uma mensagem "Holerite não encontrado".
2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC15

Breve Descrição do Caso de Uso: Consultar Peça de Estoque.

Atores: Administrativo e Motorista.

Pré-condição: Ter a peça cadastrada no banco de dados.

Fluxo Básico:

1. O administrativo ou motorista inicia o caso de uso "Consultar peça". [A1]
2. O sistema solicita nome, tipo, modelo, categoria, código de barras e descrição.
3. O administrativo ou motorista insere os dados no sistema. [E1]
4. O sistema retorna a peça especificada. [E2]
5. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo ou motorista quer cancelar o caso de uso.
2. O administrativo ou motorista clica no botão de cancelar.
3. O sistema mostra uma mensagem "Deseja realmente cancelar?".
4. O administrativo ou motorista clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro" Dados inválidos".
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um " * " .
- 3.O administrativo ou motorista insere os dados no sistema.
- 4.O administrativo ou motorista clica no botão de busca.
- 5.O sistema retorna a peça especificada.
- 6.Encerra o caso de uso.

[E2]: Peça não encontrada.

- 1.O sistema da uma mensagem" Peça não encontrado".
- 2.O sistema pede para verificar os dados corretamente.
- 3.Retoma o passo 1 do fluxo básico.

Caso de Uso: UC16

Breve Descrição do Caso de Uso: Consultar Serviço Diário.

Atores: Administrativo e Motorista.

Pré-condição: Ter o serviço cadastrado no banco de dados.

Fluxo Básico:

- 1.O administrativo ou motorista inicia o caso de uso "Consultar serviço diário".
[A1]
- 2.O sistema solicita tipo de serviço, nome do cliente, endereço, cidade, telefone, CPF/CNPJ, CEP, estado.
- 3.O administrativo ou motorista insere os dados no sistema. [E1]
- 4.O administrativo ou motorista clica no botão de busca.
- 5.O sistema retorna o serviço especificado. [E2]

6. Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

1. O administrativo ou motorista quer cancelar o caso de uso.
2. O administrativo ou motorista clica no botão de cancelar.
3. O sistema mostra uma mensagem "Deseja realmente cancelar?".
4. O administrativo ou motorista clica no botão Ok.
5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema dá uma mensagem de erro "Dados inválidos".
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um " * ".
3. O administrativo ou motorista insere os dados no sistema.
4. O administrativo ou motorista clica no botão de busca.
5. O sistema retorna o serviço especificado.
6. Encerra o caso de uso.
7. [E2]: Serviço diário não encontrada.
8. O sistema dá uma mensagem "Serviço diário não encontrado".
9. O sistema pede para verificar os dados corretamente.
10. Retoma o passo 1 do fluxo básico.

Caso de Uso: UC17
Breve Descrição do Caso de Uso: Emitir Relatório.

Atores: Administrativo e Motorista.

Pré-condição: Não há pré-condição.

Fluxo Básico:

- 1.O administrativo ou motorista inicia o caso de uso “Emitir relatório”. [A1]
- 2.O sistema solicita nome, cargo, tipo de relatório, data e descrição do relatório.
- 3.O administrativo ou motorista insere os dados no sistema. [E1]
- 4.O administrativo ou motorista clica no botão de salvar.
- 5.O sistema mostra uma mensagem “Relatório salvo com sucesso”.
- 6.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro” Dados inválidos”.
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.
- 3.O administrativo ou motorista insere os dados no sistema.
- 4.O administrativo ou motorista clica no botão de salvar.
- 5.O sistema mostra uma mensagem “Relatório salvo com sucesso”.
- 6.Encerra o caso de uso.

Caso de Uso: UC18

Breve Descrição do Caso de Uso: Realizar Serviço.

Atores: Motorista.

Pré-condição: Há uma ordem de serviço no sistema.

Fluxo Básico:

- 1.O motorista inicia o caso de uso “Realizar serviço”. [A1]
- 2.O sistema solicita nome do cliente, endereço, cidade, telefone, CPF/CNPJ, CEP, estado, e-mail, tipo do serviço, material usado no serviço, início do serviço, termino do serviço e descrição do serviço.
- 3.O motorista insere os dados no sistema. [E1]
- 4.O motorista clica no botão de salvar.
- 5.O sistema mostra uma mensagem “Dados salvo com sucesso”.
- 6.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O motorista quer cancelar o caso de uso.
- 2.O motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O motorista clica no botão Ok.
- 5.Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

- 1.O sistema da uma mensagem de erro” Dados inválidos”.
- 2.O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um “ * “.

- 3.O motorista insere os dados no sistema.
- 4.O motorista clica no botão de salvar.
- 5.O sistema mostra uma mensagem “Serviço realizado salvo com sucesso”.
- 6.Encerra o caso de uso.

Caso de Uso: UC19

Breve Descrição do Caso de Uso: Fazer Login.

Atores: Administrativo e Motorista.

Pré-condição: Não há pré-condição.

Fluxo Básico:

- 1.O administrativo ou motorista inicia o caso de uso “Fazer login”. [A1]
- 2.O sistema solicita login e senha.
- 3.O administrativo ou motorista insere os dados no sistema. [E1]
- 4.O administrativo ou motorista clica no botão logar.
- 5.O sistema mostra uma mensagem” Login efetuado com sucesso”. [E2]
- 6.Encerra o caso de uso.

Fluxos Alternativos:

[A1]: Cancelamento do caso de uso.

- 1.O administrativo ou motorista quer cancelar o caso de uso.
- 2.O administrativo ou motorista clica no botão de cancelar.
- 3.O sistema mostra uma mensagem “Deseja realmente cancelar?”.
- 4.O administrativo ou motorista clica no botão Ok.

5. Encerra o caso de uso.

Exceções:

[E1]: Dados inválidos.

1. O sistema dá uma mensagem de erro "Dados inválidos".
2. O sistema pede para informar os dados que consta como inválidos, exibindo o campo em destaque com um "*" .
3. O administrativo ou motorista insere os dados no sistema.
4. O administrativo ou motorista clica no botão logar.
5. O sistema mostra uma mensagem "Login efetuado com sucesso".
6. Encerra o caso de uso.

[E2]: Login não encontrado.

1. O sistema dá uma mensagem "Login não encontrado".
2. O sistema pede para verificar os dados corretamente.
3. Retoma o passo 1 do fluxo básico.

4. Projeto

4.1 Escolha da Metodologia

O desenvolvimento de software oferece inúmeras ferramentas para potencializar o desempenho dos profissionais envolvidos em seus processos. Devido ao prazo para desenvolvimento e entrega do produto final, foi escolhida pela equipe de desenvolvimento a metodologia Rapid Application Development (RAD) é um modelo de processo de software incremental que enfatiza um ciclo de desenvolvimento curto.

Para conseguir ênfase no desenvolvimento do projeto, foram seguidos os seguintes passos:

- Comunicação: a equipe se reuniu para entender os problemas do negócio e as características de informação que o software precisa acomodar;
- Planejamento: o planejamento foi essencial, porque foram divididas as funções em várias partes para ser trabalhada em paralelo em diferentes funções do sistema;
- Modelagem: foram feitos a modelagem do negócio usando o BPMN, modelagem dos dados com diagramas de classes, entidade-relacionamento, banco de dados, DER e modelagem do processo, com diagramas de sequências;

As fases de construção e implementação no qual faz parte da modelagem RAD será feita com a seguinte proposta:

- Construção: será enfatizado o uso de componentes e software preexistentes e a aplicação da geração automática de código.
- Implantação: estabelece a base para iterações subsequentes, se necessárias.

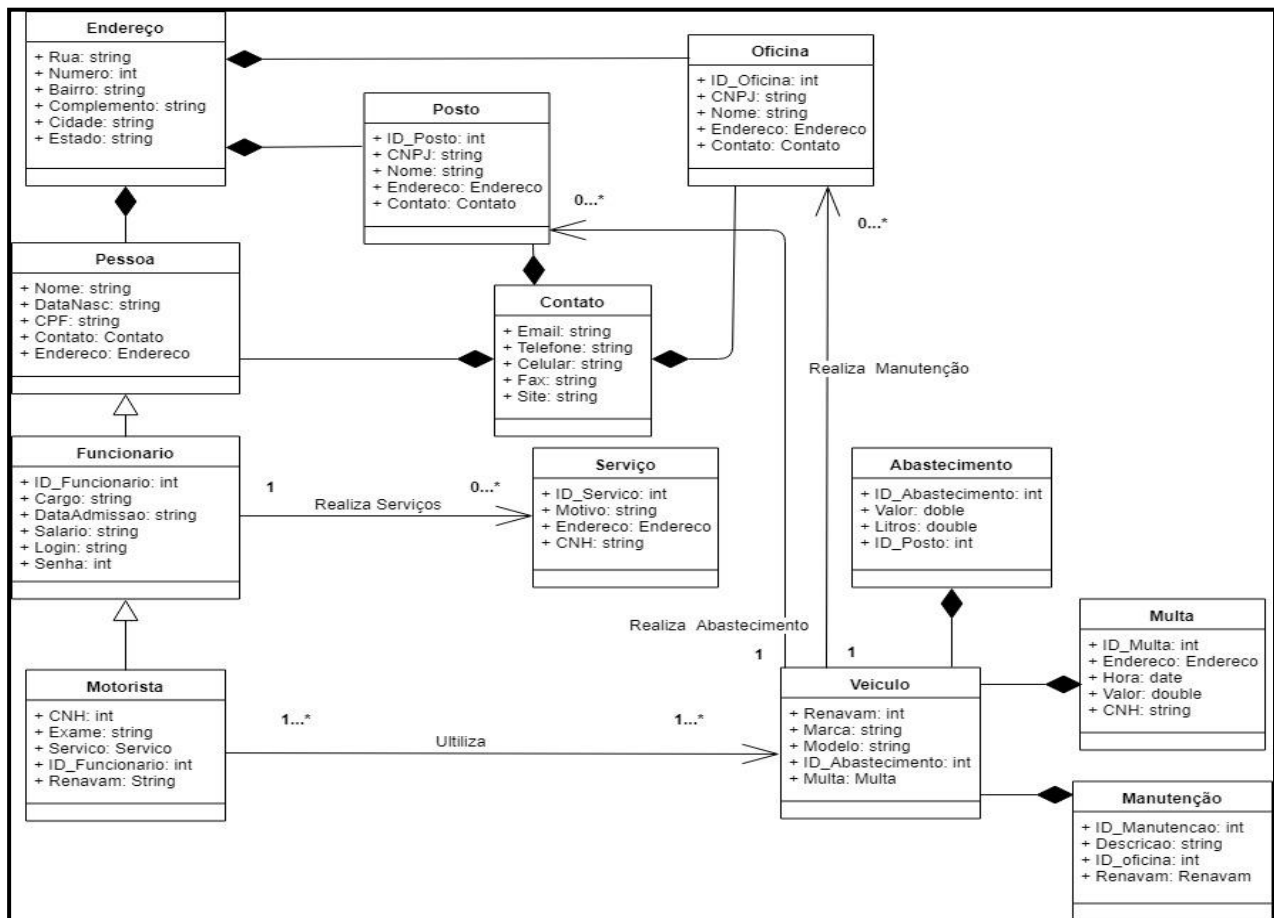
4.2 Diagramas de Classe

Ao analisar o cenário, foram identificados inúmeros objetos do sistema, contudo foi necessário fazer uma classificação dos objetos que serão utilizados, ou seja, foi necessário fazer uma Abstração para concentrarmos apenas os que eram de aspectos essenciais para o cenário do sistema, e ser criado o Diagrama de Classe. Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Pode-se afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes.

Para este sistema foi necessária a criação de várias classes e analisados os relacionamentos entre os objetos e as suas respectivas classes.

Na Figura 11 está representada o Diagrama de Classe do sistema de frota, no qual serviu de base para a elaboração do Banco de Dados do Sistema.

Figura 11 – Diagrama de Classe do Sistema de Frota



Fonte: Elaborada pelo autor. Disponível em: <<https://1drv.ms/f/s!AtCNSSxtzjrghMJZMScRO1N1qgK>>

Na Figura 11 o Diagrama de Classe foi criado usando as especificações *Unified Modeling Language* (UML) seguindo as cardinalidades e relacionamentos entre as classes. Foram criadas as seguintes classes:

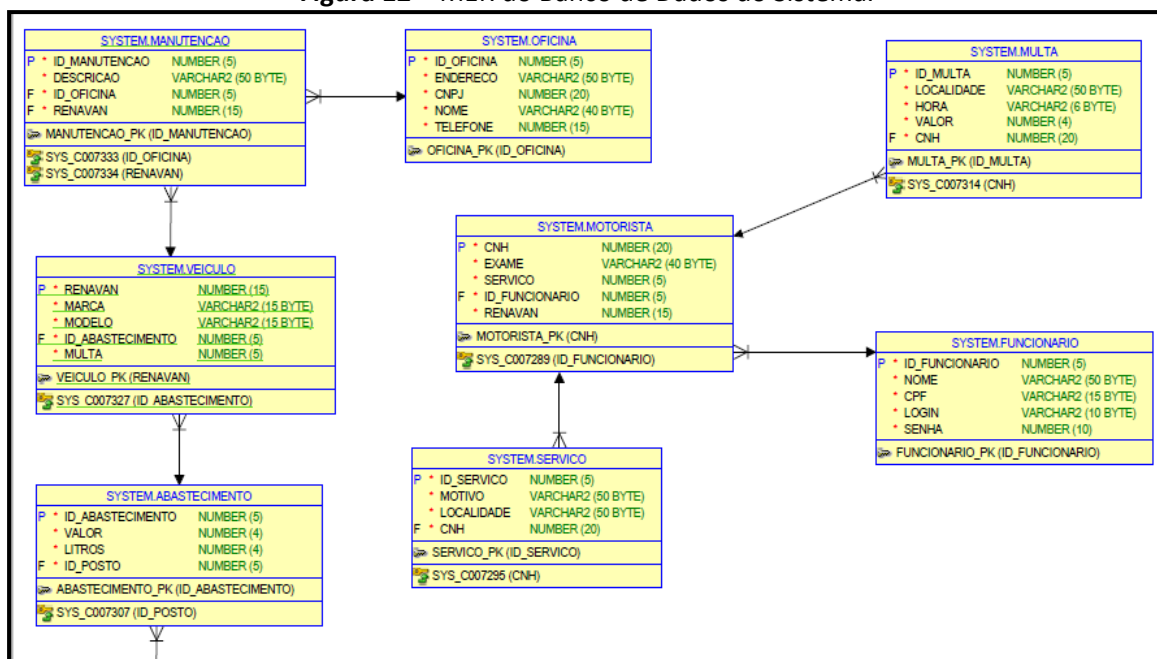
- *Pessoa*: com seus atributos comuns;
- *Funcionario*: que herda da classe *Pessoa* e tem seus atributos específicos;
- *Servico*: que tem um relacionamento com a classe *Funcionario*, onde a cardinalidade é de 1 para N;
- *Motorista*: onde a um relacionamento de agregação com *Funcionario*;

- *Veiculo*: onde tem um relacionamento com motorista, e cardinalidade de N para N;
- *Abastecimento*: onde a um relacionamento de composição com *Veiculo*;
- *Multa*: onde a um relacionamento de composição com *Veiculo*;
- *Manutenção*: onde a um relacionamento de composição com *Veiculo*;
- *Oficina*: que tem um relacionamento com a classe *Veiculo*, onde a cardinalidade de 1 para N;
- *Posto*: que tem um relacionamento com a classe *Veiculo*, onde a cardinalidade e de 1 para N;
- *Contato*: com seus atributos comuns, onde a um relacionamento de composição com as classes *Pessoa*, *Oficina* e *Posto*;
- *Endereço*: com seus atributos comuns, onde a um relacionamento de composição com as classes *Pessoa*, *Oficina* e *Posto*;

Após a criação desse diagrama de classe foi possível entender melhor o funcionamento do sistema dando base para ser criado o Banco de Dados.

Para a criação do Banco de Dados foi utilizado a ferramenta gratuita Oracle SQL Developer e após a criação das Tabelas e seus atributos, foi gerado um Diagrama chamado MER (Modelo de Entidade-Relacionamento), no qual auxiliar a melhor visualização das Classes e seus relacionamentos.

Figura 12 – MER do Banco de Dados do Sistema.



Fonte: Elaborada pelo autor. Disponível em: <<https://1drv.ms/f/s!AtCNSSxtzjrghMJZMScRO1N1qgK>>

4.3 Diagramas de Sequência

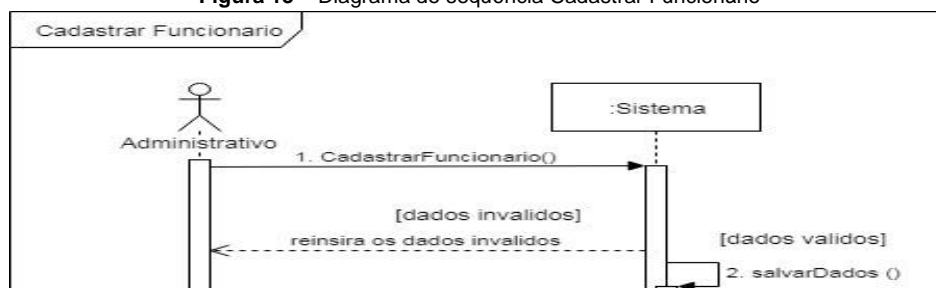
A UML – *Unified Modeling Language* ou Linguagem de Modelagem Unificada – é uma linguagem utilizada para modelar softwares baseados no paradigma de orientação a objetos, aplicada principalmente durante as fases de análise de requisitos e projeto de software. Essa linguagem consagrou-se como a linguagem-padrão de modelagem adotada internacionalmente pela indústria de Engenharia de Software, havendo um amplo mercado para profissionais que a dominem. (GUEDES, 2018)

Para entender melhor o desempenho do sistema, através das análises iniciais do documento de requisitos e estudos de casos de usos, foram desenvolvidos os Diagramas de Sequências para cada caso de uso citado neste documento no capítulo 3, seção 3.4.

O diagrama de sequência auxilia no desenvolvimento de ações de tomadas de decisões e procura determinar a sequência de eventos que ocorrem em um determinado processo, e auxiliam no entendimento do diagrama de classe que foi citado no tópico 4.2, pois os objetos representam as instancias das classes envolvidas no processo que aparecem neste diagrama estão descritas no diagrama de Classe.

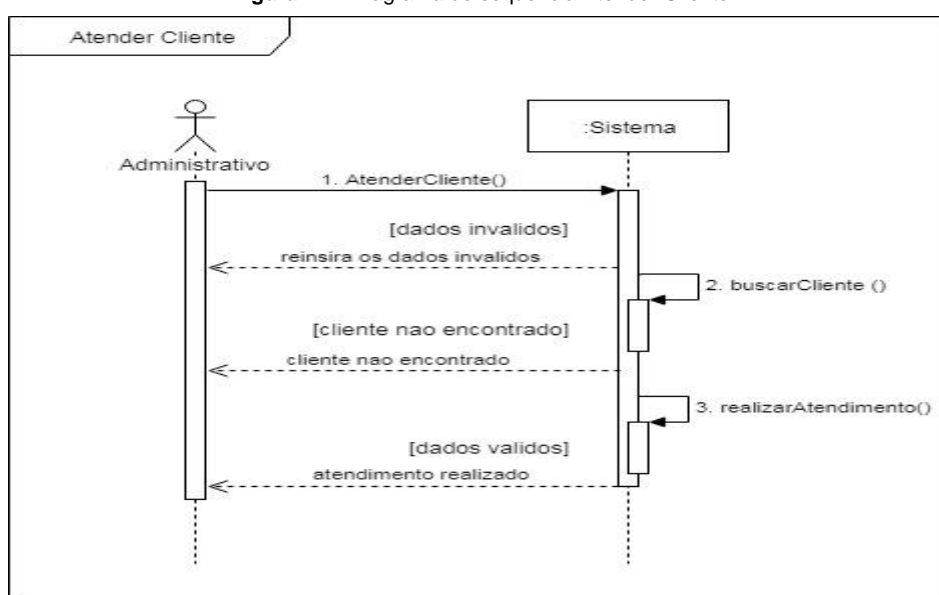
A seguir serão demonstrados dois diagramas de sequencia que foram criados para auxiliar no entendimento da sequência dos processos representados.

Figura 13 – Diagrama de sequência Cadastrar Funcionário



Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽²⁾

Figura 14 – Diagrama de sequência Atender Cliente



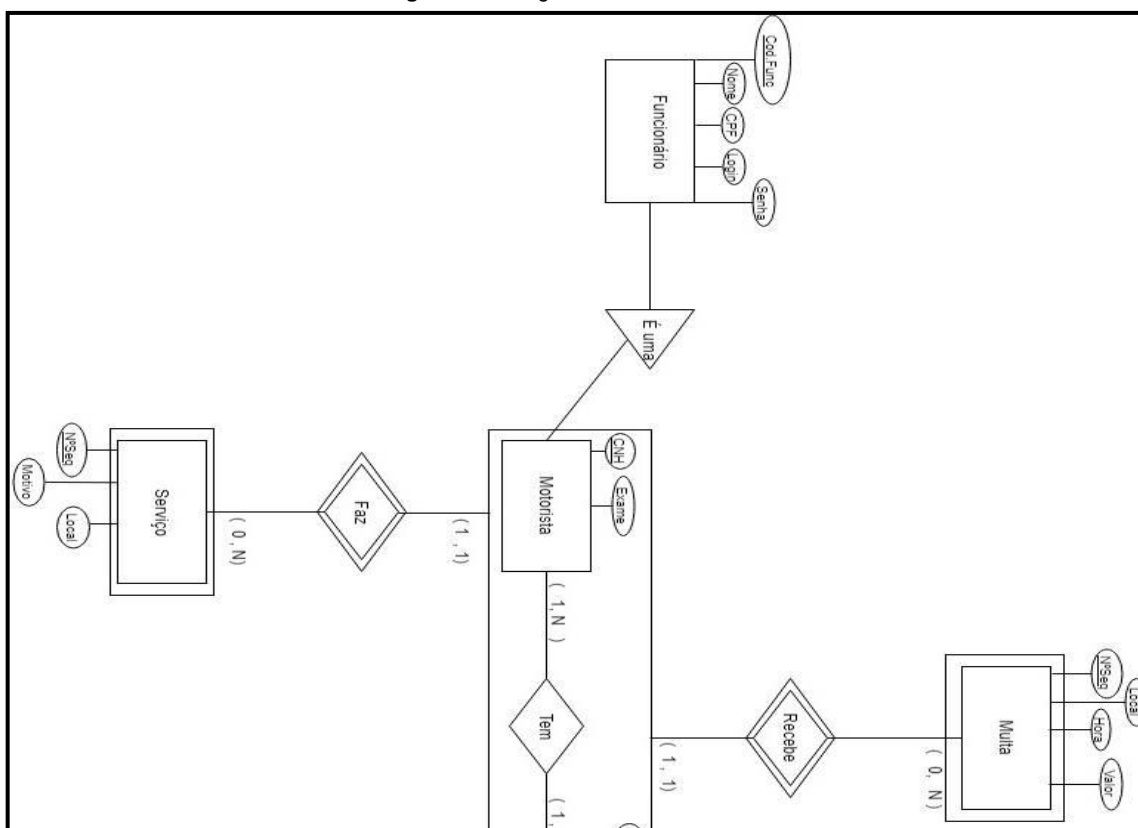
Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽¹⁾

4.4 Banco de Dados e DER

Dada a natureza do Sistema desenvolvido, é necessário implementar um banco de dados para armazenamento dos dados gerados pelo mesmo. Na implementação deste banco, inicialmente, foi elaborada a estrutura do banco, com o “diagrama entidade relacionamento”. Na Figura 15, temos a representação da seguinte situação: Uma empresa de locação de veículos chamada Frotch, possui o funcionário “motorista”, que pode usar no mínimo 1 ou no máximo N “veículos”, e o “veículo” pode possuir no mínimo 1 e no máximo N “motoristas”, o “motorista” faz um “serviço” e “serviço” pode estar associado a um “motorista”, o “motorista” pode gerar no mínimo 1 e no máximo N serviços,

este “veículo” precisa ser “abastecido”, então há um relacionamento entre o “veículo” e o “abastecimento”, o “veículo” pode abastecer no mínimo 1 vez e no máximo N vezes, exigindo um “abastecimento” em um “posto de gasolina”, o “abastecimento” pode gerar no mínimo 1 e no máximo N idas ao “posto de gasolina”, o “veículo” usado pelo funcionário pode acabar levando ou não uma “multa” então por essa razão está associado ao “motorista” e ao “veículo” (agregação), no mínimo o “motorista” e o “veículo” podem levar 0 multas, caso não tenha havido nenhuma infração de trânsito, e no máximo N multas. O “veículo” precisa de “manutenção” e por isso ele se relaciona com a “oficina” pela qual ele faz a “manutenção”, o “veículo” pode não precisar ir à oficina, caso ele não precise de manutenções, ou ir várias vezes caso ele precise de várias manutenções, portanto ele poderá ir no mínimo 0 vezes e no máximo N, resultando em idas a uma “oficina” específica, que, portanto, será a única.

Figura 15 – Diagrama Entidade-Relacionamento do Sistema



Fonte: Elaborada pelo autor. **Disponível em:** <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

5. Prototipação

5.1 Benchmark

Para entender melhor o cenário empresarial do cliente contratante, foi utilizado a técnica de Benchmark. Nessa etapa foi utilizada a técnica de comparar sites com funcionalidades similar para obter informações, e ter uma base de como iniciar a desenvolver os modelos. Nessa etapa foram definidas diretrizes de acordo com os requisitos do cliente e de acessibilidade e a paleta de cor do sistema.

Logo após essa etapa o protótipo deu-se início ao desenvolvimento com o editor *Atom*, e foi escolhida uma paleta de cores seguindo o Google Material Design⁽¹⁾, as linguagens que foram utilizadas para essa construção foram HTML, JS e CSS segue abaixo o primeiro protótipo.

Primeiro design da Tela Página inicial do sistema.

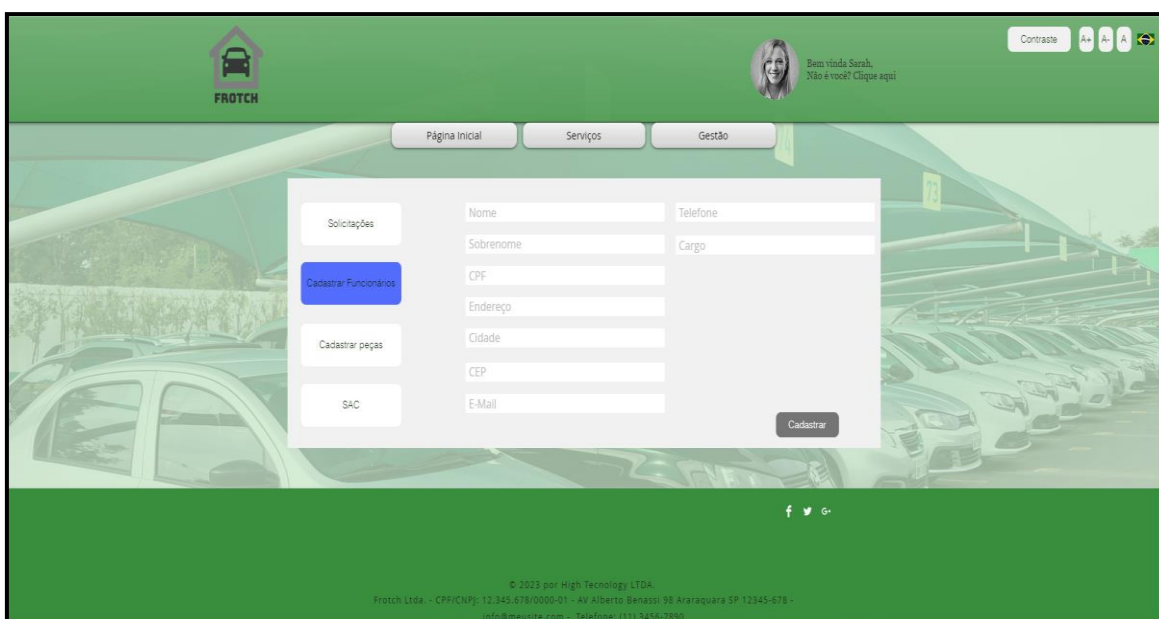
Figura 16 – Tela inicial do Sistema



Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽²⁾

Primeiro design da Tela Cadastro de funcionários.

Figura 17 – Tela Cadastro de Funcionários



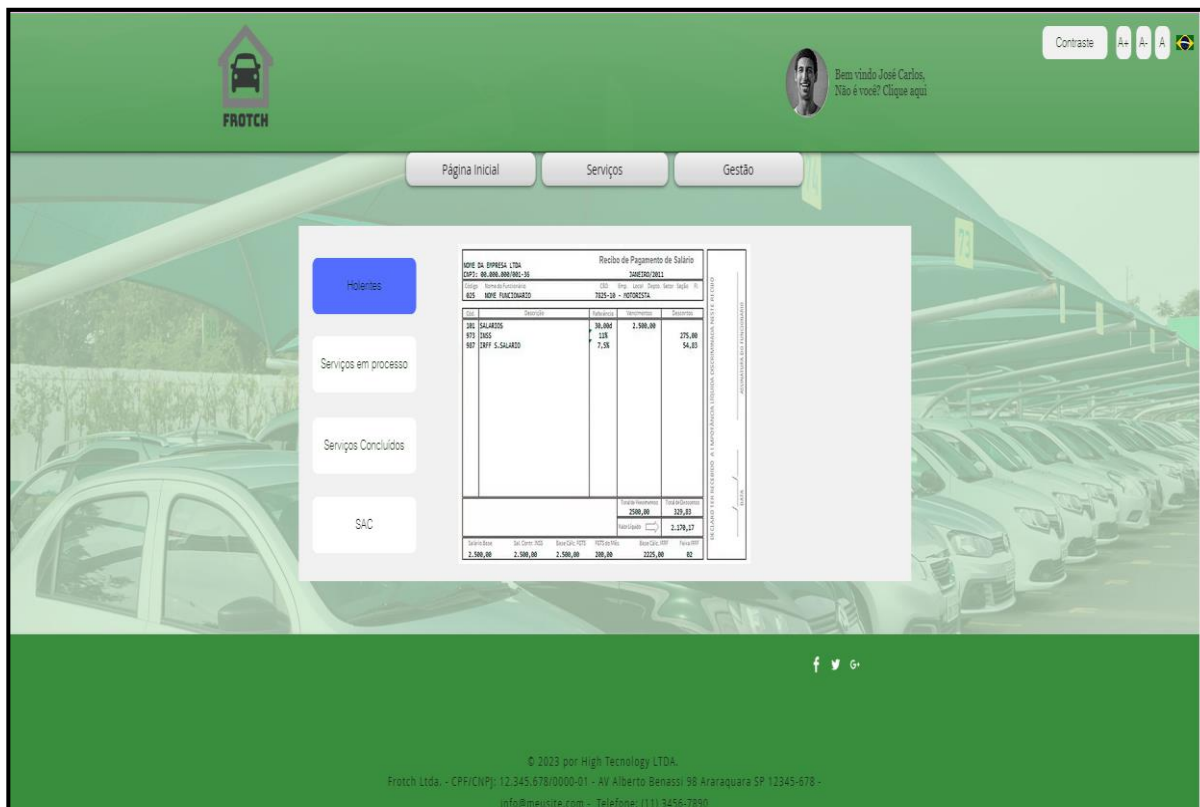
Fonte: Elaborada pelo autor. Disponível em tamanho real em: Link para acesso do OneDrive ⁽¹⁾

¹ Disponível em: Google Material Design < <https://design.google/> >

² Disponível em: <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

Primeiro design da Tela Consulta de holerites.

Figura 18 – Tela Consulta de holerites



Fonte: Elaborada pelo autor. Disponível em tamanho real em: Link para acesso do OneDrive ⁽¹⁾

Após o primeiro release das telas elas passaram pela avaliação Heurística por um grupo de desenvolvedores para verificar se as funcionalidades atendiam ao que era pedido e se estava de acordo com o termo usabilidade para que todos usuários pudessem entender e usar o sistema de maneira rápida e prática e com total liberdade.

Quadro 3. Avaliação Heurística feito por equipe externa.

Problema	Local	Heurística Violada	Severidade
Duas caixas de texto, confunde o usuário.	Busca Funcionários, Busca Peças.	9	2

Erro de grafia.	Pagina “Busca”. 3° Botão.	4	1
Informações difíceis de serem encontradas pela falta de menu	Todas as páginas	10	3
Telas com repetição de nome	Tela “sarah1 (cadastro)” e “sarah2 (cadastro)”.	4	3
Não há campos preventivos.	Telas em que necessitam cadastro.	5	2
Ajuda ao usuário.	Todas as páginas	6	1
Saída / Voltar no Sistema	Todas as páginas	3	2
Botões não estão padronizados	Baixar holerite	4	1
Como o usuário fez para chegar nessas telas? (salto de tela).	Tela “ sarah2”, “sarah1” e “busca”	1,4 e 9	4

Após essa avaliação foram verificados os erros e feitas as correções para atender aos requisitos de funcionalidade, fizemos o reuso de ideias anteriores para deixar a interface menos carregada e com funcionalidade simples e diretas.

Versão 2 da Tela Página inicial revisada com as heurísticas solucionadas.

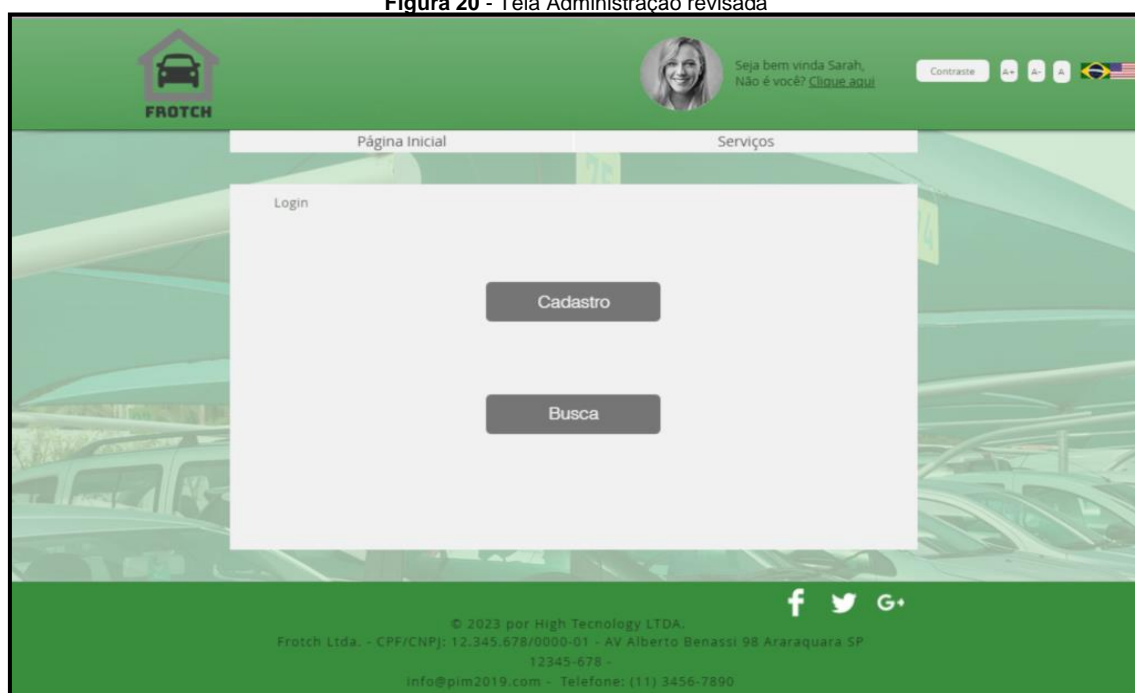
Figura 19 – Tela inicial revisada



Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽¹⁾

Versão 2 da Tela Administração revisada com as heurísticas solucionadas.

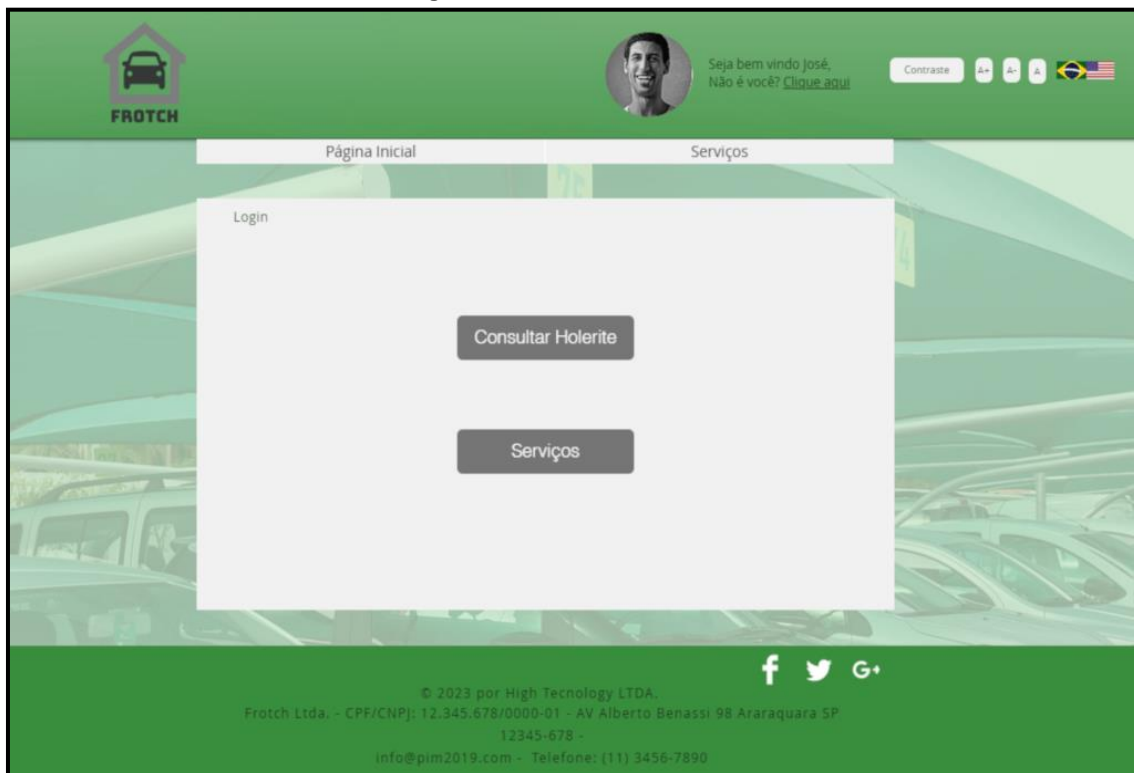
Figura 20 - Tela Administração revisada



Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽¹⁾

Versão 2 da Tela Motorista revisada com as heurísticas solucionadas.

Figura 21 - Tela Motorista revisada



Fonte: Elaborada pelo autor. Disponível em tamanho real em: [Link para acesso do OneDrive](#) ⁽¹⁾

5.2 Persona

Foram desenvolvidas para esse projeto 2 Personas, nomeadas de Sarah Lance e José Carlos onde ambos trabalham da mesma empresa de transporte, mas com cargos e funções diferentes dentro e fora do ambiente digital do sistema. Sarah cuida de toda a parte administrativa da empresa e tem acesso total a todos os conteúdos sendo eles cadastros, exclusões e edições além de controlar o estoque de peças e a frota em si, ela possui daltonismo e isso dificulta o entendimento e a percepção de cores, por isso o sistema conta com controle de contraste para se adequar corretamente e dar um grau de acessibilidade maior a Sarah.

Figura 22 - Persona representando o Administrador do Sistema



Fonte: Elaborada pelo autor. **Disponível em:** <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

Já José Carlos ele é motorista da frota, ele possui acesso ao sistema de outra maneira ele tem acesso a seus holerites, e o registro de serviços seja eles antigos e novos serviços, como a Sarah ele possui dificuldade na visão não enxerga muito bem e isso dificulta a leitura, para isso o site conta com controles de aumento e diminuição das letras para o maior entendimento e Leitura de José.

Figura 23 - Persona representando o usuário Motorista



José Carlos Gonçalves
Caminhoneiro

“Posso não ficar rico, mas sei que eu irei ser feliz”

Empresa: Empresa de transportes de carga.
Idade: 44 anos.
Genêro: Masculino.
Educação: Ensino médio.
Mídias: Usa Facebook para novas amizades, Tinder para encontrar pessoas e usa ativamente o Youtube para se distrair das longas jornadas de trabalho.
Objetivos: Tirar seu sustento, conhecer novos lugares, conhecer novas pessoas.
Desafios: Dupla jornada, Muito tempo longe de casa, Frete baixos e estradas ruins, dificuldade na visão.
Como minha empresa pode ajudá-la: Como José tem algumas dificuldades para entendimento de sistemas muitas vezes, o sistema deve ter apenas o necessário para um bom entendimento do mesmo e de outros funcionários como o acesso a rotas, datas de entrega, e locais de entrega com simples informações de uso e uma interface intuitiva

Fonte: Elaborada pelo autor. **Disponível em:** <<https://1drv.ms/f/s!AtCNSSxtzjtrghMJZMScRO1N1qgK>>

6. Conclusão

O objetivo desse trabalho foi a elaboração de um sistema que possa gerenciar todo o processo de uso da frota veicular da empresa de Telecomunicações Águias Brasil, iniciando pelo atendimento aos clientes visando seus cadastros e solicitações de serviços, tendo seu foco maior voltado ao setor de transporte, pois auxilia no gerenciamento das manutenções, abastecimento, rotas e solicitações feitas por outros setores da instituição como Almoxarifado, RH.

Com a conclusão deste projeto, ficou claro o quão importante é o mapeamento do processo de negócio da empresa para o entendimento do problema. Esse mapeamento foi ponto fundamental para que o software possa realmente atender as expectativas e solucionar os problemas do cliente. Através do mapeamento, houve um melhor levantamento dos requisitos, entendimento das ações executadas pelos usuários do sistema e qual deveria, e juntamente com o cliente, foi possível propor uma solução que satisfizesse totalmente a expectativa do cliente.

A futura implantação do software na instituição trará a possibilidade de gerar informações que auxiliaram o setor de transporte da empresa no controle de solicitações, manutenções e abastecimento, trazendo segurança e subsídios para auxiliar nas tomadas de decisões da empresa.

A elaboração do projeto possibilitou a consolidação de maneira prática daquilo que foi lecionado em sala de aula ao longo do curso de Análise e Desenvolvimento de Sistemas da Universidade Paulista Araraquara – UNIP.

7. Bibliografia:

PRESSMAN R.S.; MAXIM B. R.; - **Engenharia de Software** 8ª ed. Porto Alegre: AMGH, 2016;

SOMMERVILLE, I. - **Engenharia de Software**. 9ª ed. São Paulo: Addison Wesley, 2011;

BOOCH G., RUMBAUGH J., & Jacobson, I - **The Unified Modeling Language User Guide** 2ª ed. São Paulo: Addison Wesley Professional 2005;

GUEDES, Gilleanes. **UML Uma Abordagem Prática**. 3ª ed. Editora Novatec, 2018;

ACERVO ONLINE:

BLUEINK. **Rapid Application Development**. Disponível em: <http://www.biz/RapidApplicationDevelopment.aspx>>, Acessado em: 06 de Abril de 2019, às 09h 40.

MAGALHÃES Andréa. **[Guia Completo] Modelagem de Processos de Negócio com Notação BPMN.** Disponível em: <<https://www.dheka.com.br/ebook/guia-bpmn/>>, Acesso em: 07 de Abril de 2019, às 11h 20.

WILLIAMS Shannon. **O que é BPMN.** Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-bpmn>>, Acesso em: 07 de Abril de 2019, às 14h 05.

RABELO, Agnes. - **Analista de Marketing Rock Content** - Disponível em: <<https://rockcontent.com/blog/personas/>>, Acessado em: 01 de Maio de 2019, às 11h 20.

SANTOS, Barbara. – **Como criar persona para seu negócio** - Disponível em: <<https://blog.hotmart.com/pt-br/como-criar-persona-negocio/>>, Acessado em: 01 de Maio de 2019, às 13h 00.

SOUSA, Stella - **Significado de Benchmarking** - Disponível em: <<https://www.significados.com.br/Benchmarking/>>, Acessado em: 12 de Maio de 2019, às 07h 30.

NOVA, Thais – **Aprobio Guia Completo para 7 fontes de energias renováveis** Disponível em: <<https://aprobio.com.br/2019/01/10/um-guia-completo-para-7-fontes-de-energias-renovaveis/>>, Acessado em: 21 de Maio de 2019, às 16 h 00 .