**ANHANGUERA EDUCACIONAL**

**Unidade sorocaba**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Igor Vilalba Ferreira**

**sistema oficina mecânica**

**Sorocaba/SP**

**2020**

**Igor Vilalba Ferreira**

**sistema oficina mecânica**

Trabalho de conclusão de Projeto Integrado ao curso de Análise e Desenvolvimento de Sistemas da Anhanguera Campus Sorocaba como requisito parcial à obtenção do título de Sistema Oficia Mecânica.

Orientador: Edson Martin Feitosa

**Sorocaba/SP**

**2020**

**AGRADECIMENTOS**

Agradecemos a todos os professores da faculdade Anhanguera – Campus Sorocaba que nos auxiliaram nesse trabalho, especialmente ao professor Edson Feitosa, nosso orientador, que nos deu todo o suporte com suas correções e incentivos.

**RESUMO**

O projeto tem como objetivo de criar um sistema para uma oficina mecânica de forma que seja prático e eficiente, tanto para os funcionários como também seja ideal para os clientes. O sistema consiste em várias etapas para que funcione de forma perfeita e que não desenvolva nenhum estresse ao usuário que está utilizando do mesmo. Em alguma dessas etapas realizamos a entrevista com o cliente, podendo realizar o levantamento de requisitos (Funcionais e não funcionais), Benchmarking, Casos de usos e modelagem das entidades-relacionamentos para conclusão. Com isso conseguimos obter todos os resultados possíveis de uma forma eficaz e assertiva com os métodos que utilizamos.

**Palavras-chave:** Requisitos. Benchmarking. Casos de uso. Entidade-Relacionamento.

**Lista de Ilustrações**

[Figura 1 - Tela De Login 22](#_Toc56174808)

[Figura 2 - Tela Da Oficina Mecanica 22](#_Toc56174809)

[Figura 3 - Tela De Cadastro De Cliente 23](#_Toc56174810)

[Figura 4 - Tela De Consultar Cliente 24](#_Toc56174811)

[Figura 5 - Diagrama de Atividades 25](#_Toc56174812)

**SUMÁRIO**

[1 INTRODUÇÃO 6](#_Toc71887864)

[2 Base teórica 6](#_Toc71887865)

[2.1 Descrição do Problema 6](#_Toc71887866)

[2.2 Levantamento de Requisitos 6](#_Toc71887867)

[2.3 Benchmarking 6](#_Toc71887868)

[2.4 Caso de Uso 6](#_Toc71887869)

[2.5 Diagrama Entidade Relacionamento 6](#_Toc71887870)

[2.6 Diagrama de Caso de Uso Estendido 6](#_Toc71887871)

[2.7 Prototipagem 6](#_Toc71887872)

[2.8 Diagrama de sequência 6](#_Toc71887873)

[2.9 Diagrama de atividades 6](#_Toc71887874)

[2.10 Banco de Dados 6](#_Toc71887875)

[2.11 Conclusão do capítulo 6](#_Toc71887876)

[3 Metodologia 7](#_Toc71887877)

[3.1 Levantamento de Requisitos 7](#_Toc71887878)

[3.2 Benchmarking 7](#_Toc71887879)

[3.3 Caso de Uso 7](#_Toc71887880)

[3.4 Diagrama de Caso de Uso Estendido 7](#_Toc71887881)

[3.5 Diagrama Entidade Relacionamento 7](#_Toc71887882)

[3.6 Prototipagem 7](#_Toc71887883)

[3.6.1 Tela de Login 7](#_Toc71887884)

[3.6.2 Tela Inicial 7](#_Toc71887885)

[3.6.3 Tela Cadastro Cliente 7](#_Toc71887886)

[3.6.4 Tela Consultar Cliente 7](#_Toc71887887)

[3.7 Diagrama de sequência 8](#_Toc71887888)

[3.8 Diagrama de atividades 8](#_Toc71887889)

[3.9 Banco de Dados 8](#_Toc71887890)

[4 Conclusão 8](#_Toc71887891)

[5 Referências 8](#_Toc71887892)

[anexo a – exemplo 10](#_Toc71887893)

# INTRODUÇÃO

Este trabalho tem como o principal objetivo atender a necessidade do cliente, **Auto Mecânica Leonardo**, que possui uma oficina mecânica e tem como seu principal desafio a falta de agilidade em seus processos, gerando uma desorganização de seus clientes e serviços prestados, pois não possui nenhum tipo de sistema capacitado a atender seus requisitos. Manter o controle de dados de clientes, orçamentos, ordens de serviços e funcionários é de extrema importância para a empresa ter controle e informações precisas sobre as demandas e produtividade da oficina.

O desafio do cliente era manter as informações organizadas para trazer confiança aos clientes. Como transferir credibilidade se o cliente precisa informar os dados pessoais toda vez que necessita ir à oficina? Não possui um papel de ordem de serviço com valores, peças e informações sobre o carro? Não gerar um histórico para ajudar em futuros problemas com o carro? Essas são apenas algumas questões que causam impactos na qualidade e credibilidade com os clientes.

Com o objetivo de atender o projeto, foi inicializado uma entrevista com o cliente, onde foi questionado os processos internos da oficina, dados coletados em geral e do que o cliente necessitava. Nessa entrevista realizamos também uma das partes mais importantes do projeto, que foi o levantamento de requisitos, podendo entender e especificar corretamente o que o sistema deverá fazer, esse levantamento é fundamental para o sucesso do projeto, além de poupar tempo e dinheiro necessários para o desenvolvimento. Além do levantamento de requisitos, foi realizado também alguns outros processos como criação de casos de usos, benchmarking, diagramas de entidade-relacionamento, casos de uso extendido, prototipagem, onde conseguimos demonstrar as ideias e as características de funcionamento do sistema por meio de desenhos e por fim o banco de dados, onde está todos os dados armazenados para o sistema da oficina ser eficiente e atender a necessidade do cliente.

# Base teórica

Na base teórica vamos apresentar as pesquisas realizadas sobre os tópicos que iremos abordar nesse trabalho. Toda metodologia que implementamos no sistema, precisamos de uma base de conhecimento para podermos aplicar. Abaixo segue algumas citações sobre determinados assuntos abordados durante o trabalho.

## Descrição do Problema

A necessidade de haver um controle maior na empresa **Auto Mecânica Leonardo** é devido ao seu ramo de atividade. Dentro dos requisitos levantados através de uma entrevista com o cliente, pudemos observar que o cliente não possui nenhum tipo de controle dos clientes, orçamentos, funcionários e serviços prestados.

Levando em consideração os fatos mencionados acima, concluímos com auxílio do cliente que, para melhoria e facilidade do controle, será necessário que o sistema contenha as seguintes abas:

* Cadastrar/consultar cliente;
* Cadastrar carro;
* Cadastrar/consultar mecânico;
* Cadastrar orçamento;
* Cadastrar/consultar ordem de serviço;

Junto a este processo de implementação do sistema na empresa, vamos disponibilizar um treinamento do sistema ao usuário, e com isso, facilitamos a readaptação de controle, proporcionando maior qualidade e agilidade.

## Levantamento de Requisitos

(Sousa 2018) O levantamento de requisitos trata-se do processo de compreensão e identificação das necessidades que o cliente espera ser solucionado pelo sistema que será desenvolvido, definindo o que o software vai fazer. É a primeira etapa no ciclo de desenvolvimento de software, onde são definidos as funcionalidades e o escopo do projeto.

Existem dois tipos de requisitos que compõem um sistema: os Requisitos Funcionais (RF) e os Requisitos não-funcionais (RNF):

* **Requisitos Funcionais:** tratam das funcionalidades que o sistema deve ter.

Exemplos: cadastrar clientes, consultar saldo em conta corrente, imprimir cupom fiscal, etc.

* **Requisitos não funcionais:** estes tratam de recursos que não são funcionalidades, mas sim características do sistema, como restrições, segurança, confiabilidade, velocidade, validações, entre outros. Exemplos: velocidade (o sistema deve executar a transação em até 5 segundos), segurança (o sistema só pode ser acessado através de leitura biométrica de usuários cadastrados pelo administrador).

Para o levantamento desses requisitos existem diversas técnicas, cada uma adequada para um cenário específico, e dentre as comumente utilizadas podemos citar as seguintes técnicas:

* **Entrevista:** a forma mais utilizada, na qual o analista se reúne com o cliente e coleta os requisitos do sistema por meio de perguntas e observações do cenário apresentado pelo cliente.
* **Questionário:** o analista desenvolve um questionário e envia para o cliente responder. Através das respostas fornecidas, são elaborados os requisitos. É útil quando não é possível realizar uma entrevista pessoalmente com o cliente (embora atualmente isto é resolvido facilmente utilizando Skype, Hangouts, etc) ou quando existem diferentes usuários em locais distantes, pois pode ser enviado via e-mail. Pode ser uma boa opção para sistemas simples, porém pode se tornar inviável para sistemas mais complexos com muitos recursos e regras de negócio.

## Benchmarking

(Camargo 2019) Benchmarking vem da palavra de origem inglesa ‘benchmark’, que significa ‘referência’. Explicando melhor: isso nada mais é do que uma análise aprofundada das melhores práticas usadas por empresas do mesmo setor que o seu e que podem ser replicadas pela sua marca.

No mundo dos negócios, da produção cultural e nos esportes, a comparação é a base para a avaliação do valor daquilo que criamos ou fazemos.

Na gestão de projetos o benchmark ou benchmarking é fundamental para que você possa aprimorar a performance da sua equipe continuamente, tornando seus projetos mais eficientes e elevando a qualidade das entregas realizadas.

Ter uma metodologia própria não é garantia de que todos os processos, ações e decisões serão sempre os mais acertados. Provavelmente, deve haver outros profissionais, inclusive dentro da sua própria empresa, desenvolvendo maneiras mais efetivas de fazer aquilo que você tem dificuldades.

## Caso de Uso

(Ribeiro 2012) O caso de uso descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Os Diagramas de Casos de Uso são compostos basicamente por quatro partes:

* **Cenário:** Sequência de eventos que acontecem quando um usuário interage com o sistema.
* **Ator:** Usuário do sistema, ou melhor, um tipo de usuário.
* **Use case:** É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)
* **Comunicação:** é o que liga um ator com um caso de uso

O diagrama é composto por desenhos simples que descrevem de maneira bem objetiva o que textualmente poderia ficar extenso. Nele vemos as funcionalidades do sistema e as interações dos usuários com elas.

Existe também no caso de uso o conceito de <include> e <extend>:

* **Include:** Seria a relação de um caso de uso que para ter sua funcionalidade executada precisa chamar outro caso de uso.
* **Extend:** Esta relação significa que o caso de uso extendido vai funcionar exatamente como o caso de uso base só que alguns passos novos inseridos no caso de uso extendido.

## Diagrama Entidade Relacionamento

(Joel 2014) O Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras.

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado.

Em sua notação original, proposta por Peter Chen (idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

## Diagrama de Caso de Uso Estendido

(Oliveira 2017) A Descrição Expandida de Casos de Uso é utilizada para descrever detalhadamente o fluxo do caso de uso, quais os eventos que acontecem para garantir o sucesso do caso de uso. Algumas definições:

* Referências: Identificadores dos requisitos da tabela de requisitos que foram utilizados no caso de uso em questão.
* Pré-condições: O que é necessário que já esteja feito para que o caso de uso se inicie.
* Pós-condições: Qual será o estado do software ao ser finalizado o caso de uso.
* Requisitos Especiais: São os requisitos não funcionais, normalmente relacionados à qualidade do software. Podem estar organizados por categorias como Usabilidade, Confiabilidade, Desempenho e Manutenibilidade.
* Fluxo Básico: Evento típico para que o caso de uso seja realizado com sucesso.
* Fluxo Alternativo: Evento alternativo ao evento típico, pode ou não acontecer, podem ocorrer um ou mais eventos alternativos.

## Prototipagem

(Pereira 2019) Esta técnica tem como finalidade demonstrar as ideias e as características de funcionamento do sistema por meio de desenhos, sejam eles “rabiscos” no papel ou interfaces bem próximas do real, feitas com ferramentas que permitem esboçar a interface de uma maneira semelhante ao sistema final.

O uso de protótipos possibilita mostrar a interface, o processo de interação com funcionalidades e botões, de uma maneira fácil de se compreender. Por meio dessa informação concreta o usuário poderá não apenas entender, mas também contribuir com segurança, expressando o que gostou ou não, quais são as funcionalidades fáceis e eficazes de se utilizar etc.

**Tipos de protótipos:**

São três os tipos de protótipos, onde suas principais diferenças estão no nível de fidelidade em relação ao sistema final:

1. **Baixa-Fidelidade (wireframes)**

Esses protótipos não são semelhantes ao sistema final, na maioria das vezes, são feitos com auxílio do papel e lápis para esboçar as características iniciais da interface e o seu funcionamento. São usados também como um auxílio na conversa entre o projetista e os usuários sobre as características desejáveis e as soluções mais adequadas.

1. **Média-Fidelidade (mockups)**

Esses protótipos são mais próximos do sistema final, se comparado com os de Baixa-Fidelidade. Geralmente, são feitos utilizando ferramentas computacionais, embora não precisem ser as mesmas ferramentas que serão utilizadas para desenvolver o sistema final. Permitem simular o comportamento de interação da interface e não requerem um mesmo conhecimento técnico necessário para implementar a interface final.

1. **Alta-Fidelidade**

Esse tipo de protótipo oferece uma interface semelhante à final, pois são utilizadas as mesmas matérias (software e hardware) que serão utilizadas no sistema. Os protótipos são desenvolvidos diretamente em linguagem de programação, permitindo apresentar alguns recursos da interface com interação. Na prototipagem de Alta-Fidelidade já existe a implementação de algumas partes do sistema.

## Diagrama de sequência

O diagrama de sequência é uma solução de modelagem UML dinâmica amplamente usada porque visa especificamente linhas de vida ou processos e objetos coexistentes, bem como as mensagens trocadas entre eles para executar funções antes do final da linha de vida.

## Diagrama de atividades

(Nakazato 2013) O diagrama de atividades é uma maneira de demonstrar as ações de processos. Uma atividade é composta por uma sequência de ações. E nesta estrutura podem seguir um ou mais fluxos, que podem tomar outros caminhos através de decisões, este diagrama mostra as atividades que compõem um processo e seu fluxo.

## Diagrama de classes

Os diagramas de classes são um dos tipos mais úteis de diagramas UML, porque podem mapear claramente a estrutura de um determinado sistema ao modelar os relacionamentos entre as classes, atributos, operações e objetos de um determinado sistema.

## Banco de Dados

(Oracle 2018) Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

## Conclusão do capítulo

Levando em consideração todos os tópicos acima, concluímos que para chegarmos em uma solução ideal e completa para o cliente, precisamos entender o problema e realizar uma breve descrição, logo em seguida, realizar o levantamento de requisitos com o cliente.

Após conclusões do problema do cliente, realizamos a parte de estudo de mercado (benchmarking) e a partir dessa etapa começamos e modelar o sistema com os casos de usos e diagrama entidade-relacionamento, diagrama de caso de uso estendido, prototipagem, diagrama de atividades e o banco de dados.

# Metodologia

Após realizarmos todo o estudo sobre os tópicos necessários (descritos na base teórica), vamos implementar no sistema para finalizarmos a documentação do projeto.

Quanto a natureza da pesquisa científica utilizamos uma pesquisa aplicada, que tem como objetivo a produção de conhecimento que tenham aplicação prática e dirigida à solução de problemas reais específicos.

Na forma de abordagem do problema utilizamos a pesquisa qualitativa, pois foi identificado que várias oficinas mecânicas não estavam satisfeitas com o sistema que utilizavam, pois deixavam alguns recursos de lado e/ou custo muito alto.

Aos objetivos gerais utilizamos a exploratória, pois temos como objetivo desenvolver ideias para soluções de problemas que foram identificados durante uma investigação ampla e estuda em oficinas.

E quanto ao procedimento técnico utilizamos a pesquisa de levantamento, pois o objetivo era obtenção de informações sobre necessidades em sistemas de oficina mecânica e opiniões dos clientes.

## Levantamento de Requisitos

Levando em considerações os processos de levantamento de requisitos para compreensão da necessidade do sistema, realizamos uma entrevista com o cliente e obtivemos os seguintes resultados:

* **Fontes:** Cliente
* **Usuários:** Funcionários
* **Restrições tecnológicas:** Todas as funções não serão executadas caso o sistema operacional do cliente não suporte a versão do sistema.

**Requisito Funcional 1:**

* **Descrição:** Cadastrar / Consultar Cliente
* **Informações de entrada:** A informação de entrada para cadastro será os dados principais do cliente (Nome, CPF, RG e Telefone) e para consultar cliente somente o CPF
* **Informações de saída:** A informação de saída será um código o cliente e o cadastrado do cliente, com possibilidade de consultar dados exclusivos do cliente
* **Restrições lógicas:** Cliente já cadastrado não irá permitir um novo cadastro e CPF incorreto não permitirá a consulta

**Requisito Funcional 2:**

* **Descrição:** Cadastrar Carro
* **Informações de entrada:** A informação de entrada para cadastro será os dados principais do carro (Modelo, Ano, Cor, Marca, Placa e Código do Carro)
* **Informações de saída:** A informação de saída será um código do carro e o cadastrado completo do carro.
* **Restrições lógicas:** Carro já cadastrado não irá permitir um novo cadastro

**Requisito Funcional 3:**

* **Descrição:** Cadastrar / Consultar mecânico
* **Informações de entrada:** A informação de entrada para cadastro será os dados principais do cliente (Nome, CPF, Endereço, Telefone) e para consultar mecânico somente o CPF.
* **Informações de saída:** A informação de saída será um código do mecânico e o cadastrado completo.
* **Restrições lógicas:** Mecânico já cadastrado não irá permitir um novo cadastro

**Requisito Funcional 4:**

* **Descrição:** Cadastrar Peças
* **Informações de entrada:** A informação de entrada para cadastro será os dados principais da peça (Nome e Valor Unitário)
* **Informações de saída:** A informação de saída será um código de peça.
* **Restrições lógicas:** Todos os campos terão que ser preenchidos

**Requisito Funcional 5:**

* **Descrição:** Cadastrar / Consultar Ordem de Serviço (OS)
* **Informações de entrada:** A informação de entrada para cadastro será os dados principais dados (Dados do cliente, Dados do carro, Orçamento) e para consultar a ordem somente o código de OS.
* **Informações de saída:** A informação de saída será um código de ordem de serviço e a ficha completa.
* **Restrições lógicas:** Todos os campos terão que ser preenchidos

## Benchmarking

Após um estudo de mercado, foram encontradas algumas soluções que atenderiam a necessidade do cliente, porém com pontos positivos e negativos. Os sistemas mais relevantes disponíveis são:

**Sistema Oficina Integrada**

**Pontos positivos:**

* Emitir notas fiscais eletrônicas; Controle de estoque de peças; Gestão financeira; Cálculo de comissões.

**Pontos negativos:**

* Assinatura mensal.

**Sistema SHOficina**

Pontos positivos:

* Emitir notas fiscais eletrônicas; Agenda seus compromissos; Controle de estoque de peças.

Pontos Negativos:

* Vendido em módulos.

**Sistema Vhsys**

Pontos positivos:

* Gera orçamentos; Cotações.

Pontos Negativos:

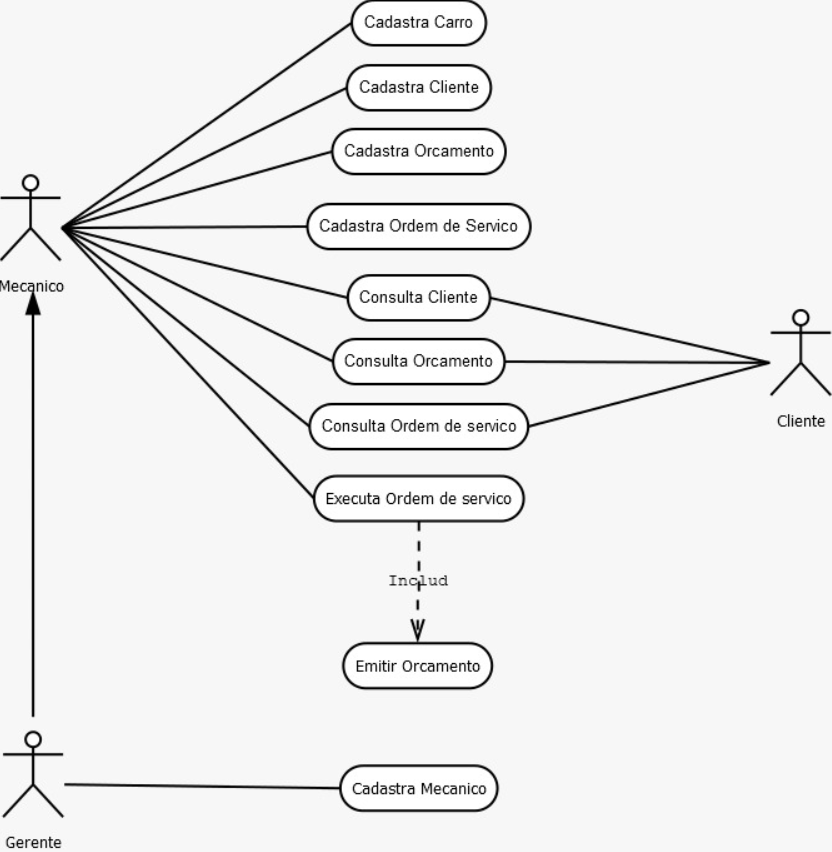
* Assinatura mensal

Sendo assim, os softwares são generalizados, portanto pouco adaptativo para todos os problemas apresentados pelo nosso cliente, cada sistema possui uma função que ajudaria na oficina, porém nenhum sistema constando todas as funções necessárias.

Levamos em consideração também o orçamento, tendo em vista que todos os sistemas mencionados acima possuem uma assinatura mensal e necessita ser utilizado em rede.

## Caso de Uso

Abaixo segue caso de uso, conforme levantamento de requisitos 3.1:



## Diagrama de Caso de Uso Estendido

**Caso de uso: Cadastrar Ordem de Serviço**

1. [IN] O Usuário (mecânico) abre uma nova ordem de serviço;
2. [OUT] O sistema pede o Nome ou ID do cliente;
3. [IN] O Usuário insere Nome ou ID do cliente;
4. [OUT] O Sistema Pede para selecionar um carro cadastrado do cliente;
5. [IN] O Usuário seleciona um carro para a Ordem de Serviço;
6. [OUT] O Sistema Pede para o Usuário selecionar as peças;
7. [IN] O Usuário Seleciona as peças;
8. [OUT] O Sistema pede para inserir o valor da mão de obra;
9. [IN] O Usuário insere o valor da mão de obra;
10. [OUT] O Sistema gera uma ordem de serviço com os dados fornecidos;
11. [IN] O Usuário Salva A Ordem de Serviço;

**Exceção:**

Exceção 3ª: Cliente não cadastrado;

3a.1 [IN] O comprador informa seu CPF, nome, endereço telefone;

Retornar ao passo 3;

Exceção 4ª: Carro não cadastrado;

4a.1 [IN] O Usuário realiza o cadastro do carro;

Retornar ao passo 4;

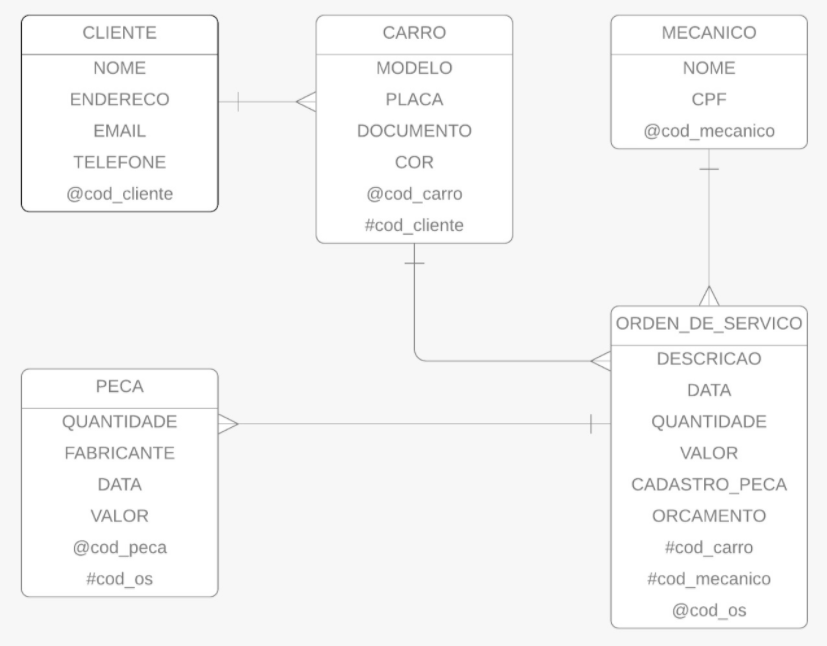
Exceção 6ª: Peça não cadastrada;

6a.1 [IN] O Usuário realiza o cadastro da peça;

Retornar ao passo 7;

## Diagrama Entidade Relacionamento

Abaixo segue o modelo de DER realizado, conforme levantamento de requisitos:

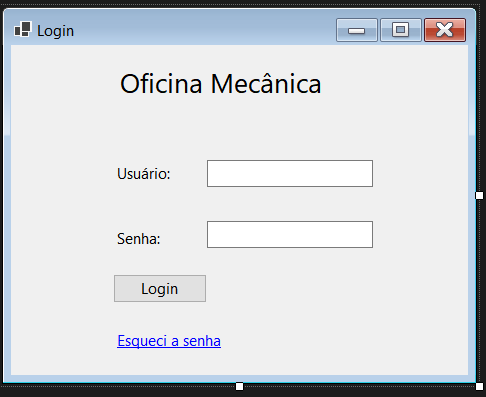


## Prototipagem

### Tela de Login

Nessa primeira tela será a tela que o usuário deverá fornecer seu login (CPF) e senha, de acordo com o cadastro realizado. Somente administradores vão ter acesso.

Figura 1 - Tela De Login



Fonte: Próprio Autor

### Tela Inicial

Após efetuar o login, o usuário será direcionado para a tela inicial, onde terá todas as opções de funções do sistema.

Figura 2 - Tela Da Oficina Mecânica



Fonte: Próprio Autor

### Tela Cadastro Cliente

Após o usuário realizar o login e ir para tela inicial, terá a opção de cadastrar os clientes na opção “Cadastre Cliente”.

Nessa tela será informado os dados pessoais do cliente para futuras consultas e ordens de serviços.

Figura 3 - Tela De Cadastro De Cliente

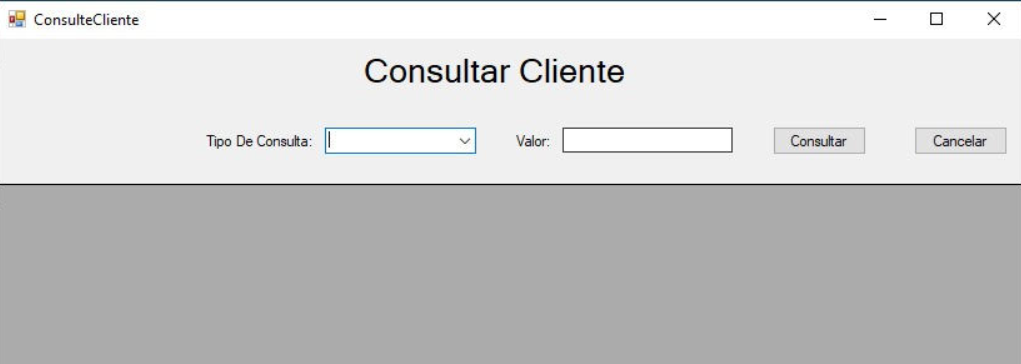


Fonte: Próprio Autor

### Tela Consultar Cliente

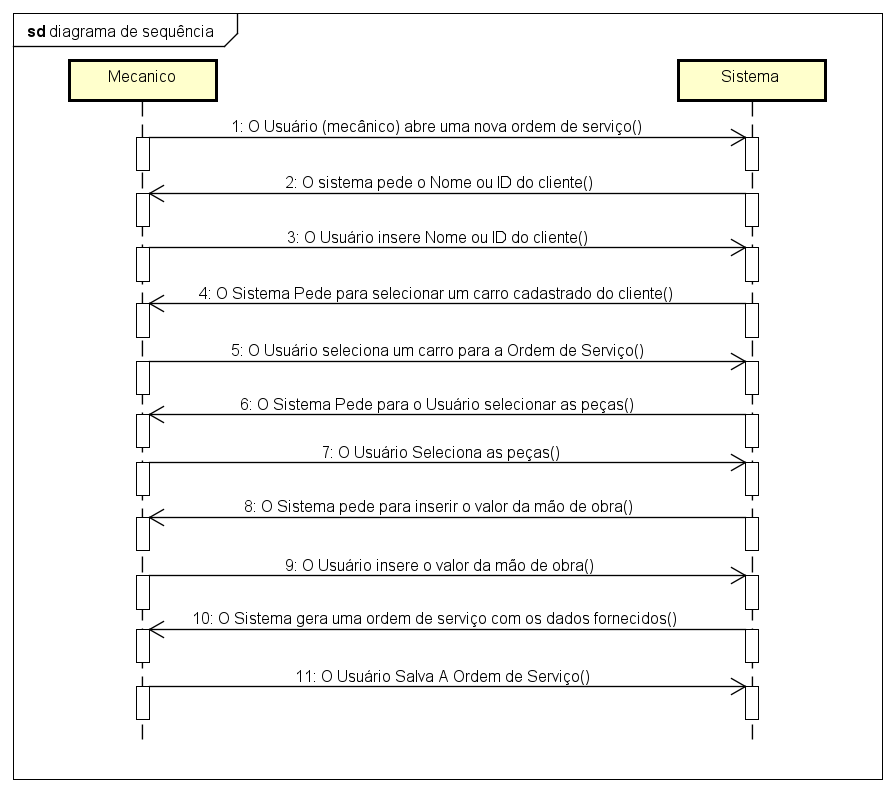
Após ter o cliente cadastrado, o usuário poderá pesquisar alguns dados sobre o cliente através do CPF.

Figura 4 - Tela De Consultar Cliente



Fonte: Próprio Autor

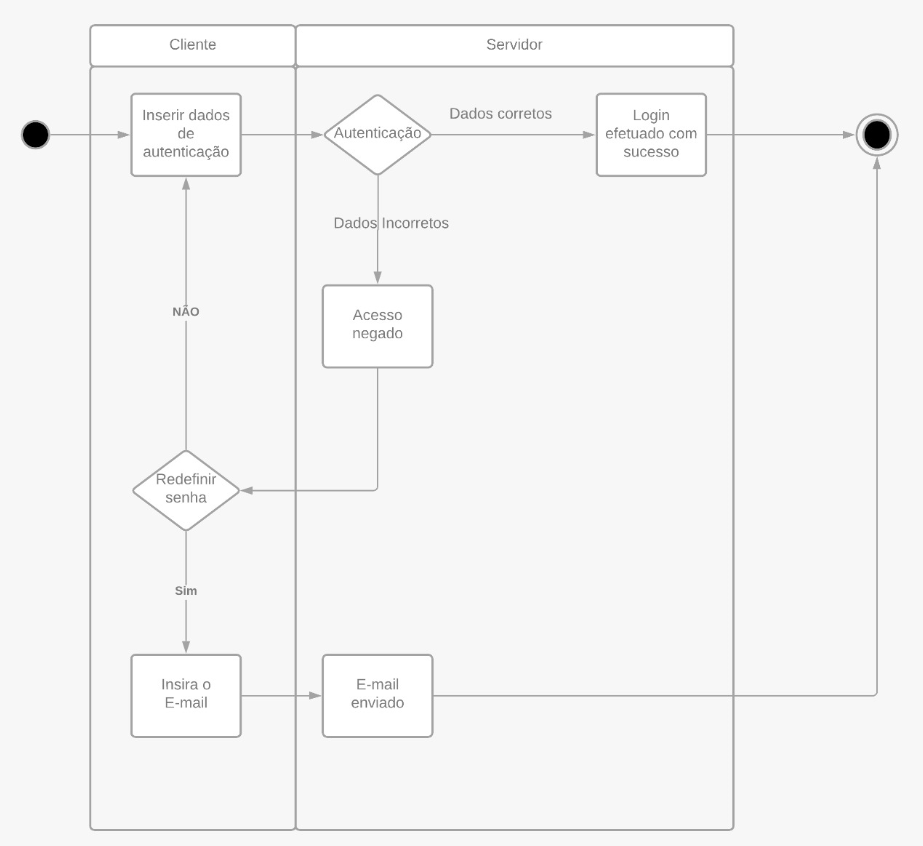
## Diagrama de sequência



## Diagrama de atividades

O diagrama de atividades é um gráfico de fluxo que nos ajuda a entender as sequencias de um processo computacional. No caso do programa para a oficina mecânica, os dados de autenticação são inseridos pelo cliente, e, caso os dados sejam corretos e autenticados pelo servidor, o login é efetuado. Caso os dados sejam incorretos, o acesso é negado, e é gerada a opção de redefinir senha, com a possibilidade de outra tentativa de inserção de dados corretos, ou opção de envio de e-mail para correção de senha, encerrando o processo.

Figura 5 - Diagrama de Atividades



Fonte: Próprio Autor

## Diagrama de classes (Atualizar)

XXXX

## Classes (Codificação) (Atualizar)

xxxx

## Banco de Dados

Abaixo o script utilizado para criação do banco de dados da oficina mecânica, onde nos baseamos no diagrama entidade relacionamento (3.5).

*create database mecanica;*

*use mecanica;*

*create table cliente(*

*cod\_cliente int primary key auto\_increment,*

*nome varchar(30)not null,*

*cpf varchar(11) not null,*

*endereco varchar(500) not null,*

*email varchar(50) null,*

*telefone varchar(30) not null*

*);*

*create table carro(*

*cod\_carro int primary key auto\_increment,*

*modelo varchar(100) not null,*

*placa varchar(10) not null,*

*documento varchar(1000),*

*cod\_cliente int references cliente(cod\_cliente)*

*);*

*create table mecanico(*

*cod\_mecanico int primary key auto\_increment,*

*nome varchar(50) not null,*

*cpf varchar(11) not null*

*);*

*create table ordem\_servico(*

*cod\_os int primary key auto\_increment,*

*descricao\_produto varchar(5000)null,*

*datta datetime not null,*

*quantidade int,*

*valor numeric(10,2)not null,*

*cadastro\_peca varchar(5000),*

*orcamento varchar(5000),*

*cod\_carro int not null references carro(cod\_carro),*

*cod\_mecanico int not null references mecanico(cod\_mecanico)*

*);*

*create table peca(*

*cod\_peca int primary key auto\_increment,*

*quantidade int not null,*

*fabricante varchar(100) not null,*

*data datetime not null,*

*valor numeric(10,2) not null,*

*cod\_os int not null references ordem\_servico(cod\_os)*

*);*

# Conclusão

Podemos concluir que o intuito desse desenvolvimento possibilitou uma análise de como um software foi feito para atender as necessidades de uma mecânica.

Foram elaborados processos de caso de uso junto a um diagrama de entidade de relacionamento (DER), diagrama de atividades e diagrama de caso de uso extendido, para mostrar de como seria a estrutura do sistema e também levantados pontos positivos, negativos e requisitos funcionais e não funcionais, e assim foi elaborado um benchmarking para análise e estudo de outras práticas.

Após esse processo de definição e estrutura das funções do sistema, foi elaborado a protipagem, onde o objetivo foi demonstrar as ideias e as características de funcionamento do sistema por meio de desenhos e com aprovação do cliente foi construído o banco de dados da oficina mecânica.

O sistema pode melhorar a apuração dos resultados na falta de agilidade e organização dentro da empresa em si. Nesse sentido, a utilização de um sistema próprio permite aos funcionários realizarem seu trabalho de forma mais rápida e eficiente, além disso, diminuindo o tempo de espera dos clientes.

# Referências

Camargo, Robson. *RC Robson Camargo – Projetos e Negócios.* 19 de Fevereiro de 2019. https://robsoncamargo.com.br/blog/Benchmarking (acesso em 22 de 04 de 2020).

Joel. *DEVMEDIA.* 04 de Janeiro de 2014. https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332 (acesso em 06 de Maio de 2020).

Nakazato, Karen Miyuki. *TÉCNICA DE LEITURA PARA INSPEÇÃO DE DIAGRAMAS DE ESTADOS COM.* 2013.

Oliveira, Mônica. *https://treinamentowaei.wordpress.com/descricao-expandida-do-caso-de-uso/.* janeiro de 2017.

*Oracle.* 2018. https://www.oracle.com/br/database/what-is-database.html.

Pereira, Ricardo. “Medium.” 2019. https://medium.com/contexto-delimitado/prototipagem-de-software-7ac07027e6d8.

Ribeiro, Leandro. *DEVMEDIA.* 2012. https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408 (acesso em 28 de Abril de 2020).

Sousa, Julio Martins de. *Cedrotech.* 31 de Outubro de 2018. https://blog.cedrotech.com/levantamento-de-requisitos-o-ponto-de-partida-do-projeto-de-software/ (acesso em 22 de 04 de 2020).

anexo a – exemplo