

Hoelesen França do Nascimento

# **Sistema de controle de lavagens de veículos**

Vilhena - RO

2022



Hoelesen França do Nascimento

## **Sistema de controle de lavagens de veículos**

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – *campus* Vilhena, realizado em cumprimento de requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO

*Campus* Vilhena

Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

Orientador: Gilberto Pereira da Silva

Vilhena - RO

2022

N244s NASCIMENTO, Hoelesen França do

Sistema de controle de lavagens de veículos/ Hoelesen França do Nascimento. – Vilhena/RO, 2022.

67f. il.:

Orientador: Prof. Gilberto Pereira da Silva.

Monografia (Curso Tecnólogo em Análise e desenvolvimento de sistemas) - Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO.


1. Microempreendedores 2. Gerência 3. Software I. Título.

CDD 005.133

Hoelesen França do Nascimento


## Sistema de controle de lavagens de veículos

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – *campus* Vilhena, realizado em cumprimento de requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Documento assinado digitalmente  
 GILBERTO PEREIRA DA SILVA  
Data: 13/12/2022 12:34:48-0300  
Verifique em <https://verificador.iti.br>


---

Msc. Gilberto Pereira da Silva  
Orientador

Documento assinado digitalmente  
 ROSA MARIA DA SILVA GONCALVES  
Data: 13/12/2022 17:46:40-0300  
Verifique em <https://verificador.iti.br>


---

Msc. Rosa Maria da Silva Goncalves

Documento assinado digitalmente  
 ERICK LEONARDO WEIL  
Data: 13/12/2022 17:52:40-0300  
Verifique em <https://verificador.iti.br>

---

Esp. Erick Leonardo Weil

Documento assinado digitalmente  
 WEIDE CASSIMIRO JERONIMO  
Data: 13/12/2022 17:14:36-0300  
Verifique em <https://verificador.iti.br>

---

Esp. Weide Cassimiro Jeronimo

Vilhena - RO  
2022



*Dedico este trabalho à minha família  
e, principalmente, aos mestres que sempre mostraram que o aprender é a melhor opção!*



# Agradecimentos

Aos meus familiares, por todo o apoio e ajuda bem como por muito contribuírem para a realização deste trabalho.

Aos meus colegas de turma, por compartilharem comigo tantos momentos de descobertas e aprendizado como também por todo o companheirismo, ao longo deste percurso.

Aos professores, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.



# Resumo

As pequenas empresas, nos últimos anos, apresentaram um grande crescimento, mesmo em um momento de crise econômica no país. O segmento de prestação de serviço em lavagem de veículos, mais conhecido como lava-jato, acompanhou essa evolução. Dentre essas pequenas empresas, muitas não possuem um sistema de gerência e controle de serviços, sendo necessário administrar todo o fluxo da empresa de maneira física e manual. Após o entendimento desse problema, foi elaborado uma pesquisa com alguns microempreendedores. A partir dos resultados originou-se o projeto de um sistema de controle de lavagem de veículos voltado somente às empresas e não ao público a qual elas prestam serviços. O objetivo deste trabalho é criar uma solução plausível ao problema existente. Dessa forma, como trabalho de conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistema, foi desenvolvido um software com funcionalidades que permitem aos profissionais do ramo de lavagens trabalharem de uma maneira mais eficiente.

**Palavras-chave:** microempreendedores; gerência; eficiente; software.



# Abstract

Small companies, in recent years, have shown great growth, even at a time of economic crisis in the country. The vehicle wash service segment, better known as car wash, followed this evolution. Among these small companies, many do not have a service management and control system, and it is necessary to manage the entire flow of the company physically and manually. After understanding this problem, a survey was carried out with some microentrepreneurs. From the results, the project of a vehicle wash control system was created, aimed only at companies and not at the public to which they provide services. The objective of this work is to create a plausible solution to the existing problem. Thus, as a conclusion work for the Technology Course in System Analysis and Development, a software was developed with functionalities that allow professionals in the field of washing to work more efficiently. Keywords: microentrepreneurs; management; efficient; software.

**Keywords:** microentrepreneur; management; efficient; software.



# Lista de ilustrações

Figura 1 – Processos . . . . .	30
Figura 2 – Fase de iniciação . . . . .	31
Figura 3 – Fase de execução . . . . .	32
Figura 4 – Fase de finalização . . . . .	33
Figura 5 – Diagrama de classe . . . . .	34
Figura 6 – Modelagem de dados . . . . .	35
Figura 7 – Testes Lighthouse . . . . .	37
Figura 8 – Arquitetura MVC . . . . .	38
Figura 9 – Arquivo README.md . . . . .	44
Figura 10 – Tela de login . . . . .	45
Figura 11 – Tela inicial . . . . .	46
Figura 12 – Menu cadastros . . . . .	47
Figura 13 – Cadastro de cliente . . . . .	48
Figura 14 – Cadastro de veículo . . . . .	48
Figura 15 – Tipo de serviço . . . . .	49
Figura 16 – Adicionar serviço . . . . .	50
Figura 17 – Todos serviços . . . . .	51
Figura 18 – Financeiro . . . . .	52
Figura 19 – Tela de login software . . . . .	59
Figura 20 – Tela inicial . . . . .	60
Figura 21 – Tela de cadastros . . . . .	61
Figura 22 – Todos os serviços . . . . .	62
Figura 23 – Tela financeiro . . . . .	63



# Lista de abreviaturas e siglas

SGBD - Data Base Management System

IOS - Sistema operacional movel Apple

APP - Aplicativo

BPNM - Business Process Model and Notation

UML - Unified Modeling Language

ORM - Object Relational Mapper

CSS - Cascading Style Sheets

MVC - Model View Controller

MVP - Minimum Viable Product

DB - Data Base

MIT - Massachusetts Institute of Technology

API - Application Programming Interface

CSV - Extensão de arquivo, Comma Separated Values



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Contexto e problema</b>	<b>19</b>
<b>1.2</b>	<b>Objetivos</b>	<b>19</b>
1.2.1	Objetivo geral	19
1.2.2	Objetivos específicos	20
<b>1.3</b>	<b>Justificativa</b>	<b>20</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>21</b>
<b>2.1</b>	<b>O que é software</b>	<b>21</b>
<b>2.2</b>	<b>Como o software ajuda na sociedade</b>	<b>21</b>
<b>2.3</b>	<b>Qualidade de software</b>	<b>22</b>
<b>2.4</b>	<b>Linguagem de programação Python</b>	<b>22</b>
2.4.1	Ambiente de execução	23
<b>2.5</b>	<b>Banco de dados</b>	<b>23</b>
<b>2.6</b>	<b>Versionamento</b>	<b>24</b>
<b>2.7</b>	<b>Trabalhos similares</b>	<b>25</b>
2.7.1	Software proprietário	25
2.7.2	Software código aberto	26
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>27</b>
<b>3.1</b>	<b>Materiais - Ferramentas e tecnologias utilizadas</b>	<b>27</b>
3.1.1	HEFLO	27
3.1.2	Visual Studio Code	27
3.1.3	Git e Github	27
3.1.4	UML	27
3.1.5	Python	28
3.1.6	Django	28
3.1.7	Bootstrap	28
3.1.8	Persistência de dados com SQLite	28
<b>3.2</b>	<b>Métodos</b>	<b>29</b>
3.2.1	Ciclo de vida e processos	30
3.2.2	Fase de iniciação	31
3.2.3	Fase de execução	32
3.2.4	Fase de finalização	32
3.2.5	Controle de Versão	33
3.2.6	Modelagem	33

---

<b>3.3</b>	<b>Requisitos</b> . . . . .	<b>35</b>
3.3.1	Requisitos funcionais . . . . .	35
3.3.2	Requisitos não funcionais . . . . .	36
<b>3.4</b>	<b>Arquitetura do software</b> . . . . .	<b>37</b>
<b>3.5</b>	<b>Licença de uso</b> . . . . .	<b>39</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b> . . . . .	<b>41</b>
<b>4.1</b>	<b>Gerenciamento de configuração e mudanças</b> . . . . .	<b>41</b>
<b>4.2</b>	<b>Processo de desenvolvimento</b> . . . . .	<b>41</b>
<b>4.3</b>	<b>Documentação</b> . . . . .	<b>42</b>
4.3.1	Documentação para usuários . . . . .	42
4.3.2	Documentação para desenvolvedores . . . . .	42
<b>4.4</b>	<b>Demonstração do software</b> . . . . .	<b>45</b>
4.4.1	Login . . . . .	45
4.4.2	Cadastros . . . . .	47
4.4.3	Serviços . . . . .	50
4.4.4	Financeiro . . . . .	52
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>53</b>
<b>5.1</b>	<b>Trabalhos futuros</b> . . . . .	<b>53</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>55</b>
	<b>APÊNDICES</b> . . . . .	<b>57</b>
	<b>APÊNDICE A – DOCUMENTAÇÃO PARA USUÁRIOS</b> . . . . .	<b>59</b>
	<b>ANEXOS</b> . . . . .	<b>65</b>
	<b>ANEXO A – LICENÇA MIT</b> . . . . .	<b>67</b>

# 1 Introdução

## 1.1 Contexto e problema

Atualmente, as empresas do ramo de lavagem de carros recebem os veículos e as informações como, nome do cliente, placa, serviço a ser prestado, são registradas pelo gerente em uma planilha física. Esses dados registrados são utilizados para controlar melhor os serviços a serem realizados, horários de chegada e saída bem como agendamentos. Ao final do mês, os dados levantados são todos listados para o balanço financeiro.

A cidade de Vilhena possui uma vasta quantidade de empresas do ramo de lavagem de veículos que apresentam amplos serviços além dos habituais de limpeza. Mesmo com essa quantidade significativa de empresas, o sistema de gerência mostra-se falho por muitas vezes não conseguir acompanhar inovações tecnológicas. Um dos fatores que pode dificultar essa evolução relaciona-se aos poucos softwares, disponíveis no mercado, com funcionalidades complexas bem como com muitas informações que geram dificuldades para quem as utiliza. Dito isso, a criação de um software para suprir essa demanda mostra-se necessário.

A interface do software deve ser web, abrangendo uma enorme quantidade de dispositivos capazes de processar as informações, sem necessidade de um processamento pesado na ponta, o sistema todo é processado em um servidor e entregue ao usuário a fim de que facilite e agilize todo o processo.

De maneira resumida, este software utiliza da linguagem python junto ao *framework* Django, criando uma aplicação agradável. Essa solução tem como fim gerenciar os serviços básicos utilizados nas empresas de lavagem de veículos, viabilizando a melhorias para a gestão interna atualizando e automatizando serviços para o meio digital.

Objetiva-se, por meio deste processo de desenvolvimento, a criação de uma aplicação de gestão funcional, com o interesse geral de organizar tarefas internas de empresas de lavagem de veículos.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

Desenvolver um sistema WEB para gerenciar a prestação de serviço de uma empresa de lavagem de veículos.

### 1.2.2 Objetivos específicos

- Entrevistar empresa Lava Jato Cristal para montar o plano de desenvolvimento do projeto.
- Elicitar requisitos funcionais e não funcionais.
- Analisar os requisitos levantados, identificando características para o sistema.
- Escolher as tecnologias a serem utilizadas no desenvolvimento do projeto WEB.
- Implantar o projeto em ambiente de produção na empresa Lava Jato Cristal.

## 1.3 Justificativa

O processo de criação de uma aplicação é um aprendizado tanto à empresa quanto ao desenvolvedor. Dentro do processo serão pautadas as necessidades que não foram supridas por outros softwares do ramo. O sistema será de grande utilidade ao mercado, tendo em vista os poucos softwares gratuitos ao público, voltados especificamente para a área desse segmento. O sistema vai automatizar o processo de armazenamento de dados, gerando agilidade na prestação de cada serviço, permitindo estabelecer um plano de crescimento do negócio de lavagens de veículos com base em dados dos relatórios de serviço e financeiros gerados pelo sistema.

## 2 Fundamentação teórica

Neste ponto objetiva-se discutir algumas informações obtidas por meio de pesquisa bibliográfica e, assim, contribuir com o processo de desenvolvimento de softwares. Buscando desenvolver o crescimento da tecnologia em questões gerais, e a influência da mesma no processo de comunicação e entendimento da sociedade, com questões que se revelam tendo como base experiência passadas. Assim, analisando os principais temas de referência e definição de conceitos, dimensionar elementos abordados, bem como os modelos e demandas relacionados a criação e utilização de softwares. Em suma, tem-se o campo de conhecimento como os principais temas de referência e definição de conceitos, a fim de estimar os elementos abordados bem como os modelos e demandas relacionados à criação e à utilização de softwares.

### 2.1 O que é software

Segundo Sommerville (2007, p.20), software é caracterizado como um programa de computador e toda a bagagem de documentação a ele associado. Acompanhando a história, após o surgimento dos computadores, os softwares eram desenvolvidos sob medida para cada empresa que por fim iria utilizá-lo mantendo em sigilo seus métodos de desenvolvimento sendo eles conhecidos apenas pelos desenvolvedores, os desenvolvimentos eram elaborados para funcionar nas plataformas específicas das empresas levando em consideração os processos de trabalho já estipulados. Após anos e muitas mudanças, o software na atualidade é mais flexível no seu desenvolvimento e pode abranger diversas plataformas mais parametrizáveis possível.

### 2.2 Como o software ajuda na sociedade

Decorrendo o pensamento de Sommerville “O mundo moderno não poderia existir sem o Software” (2007, p.20), o pesquisador usa essas palavras referindo-se a sistemas que hoje atuam na qualidade de vida dos usuários desse serviço computacional. Pode se afirmar que o software não é somente usado industrialmente, parte de um sistema que engloba várias áreas como sistemas de saúde, entretenimento, a indústria global da música, jogos de computador, cinema, televisão, e o sistema financeiro que dominam o funcionamento e interligação entre instituições bancárias de países.

## 2.3 Qualidade de software

De acordo com, Beckman (1999, p.35) “gerir o conhecimento é formalizar as experiências vividas, conhecimentos e saberes tanto do indivíduo e de todo o coletivo”, ao partir desse ponto de vista, o autor passa a ideia que documentar a experiência é a melhor maneira de estimular o progresso de um grupo, repassando o acesso às mesmas, a fim de formar indivíduos mais competentes para não ter que passar pelas mesmas dificuldades e assim alcançar um patamar superior.

Ao abrir um comparativo diretamente ao software, toda a história vai criando uma nova geração de produtos mais atuais e também de maior qualidade. Embora o conceito de qualidade não seja muito conhecido pela grande maioria, e o autor Côrtes (2001, p. 15) representa listando algumas definições sobre a qualidade de software:

- O software não pode possuir defeitos.
- Deve ter sua finalidade para o uso adequado.
- Deve atender as especificações a qual se propôs.
- Possuir as características de confiabilidade / usabilidade / manutenibilidade.

Gestores com diferentes interesses em um produto tem diferentes visões sobre funcionamento e a qualidade de um software.

## 2.4 Linguagem de programação Python

As linguagens de programação são essenciais na criação de novos softwares e sites como também são muito utilizadas em áreas de maior complexidade como sistemas de inteligência artificial e ciência de dados, por exemplo.

Python é considerada uma linguagem de programação de alto nível. Essa definição aparece devido ao fato de possuir tais características, ser dinâmica interpretada, orientada a objetos, multiplataformas e modular.

A linguagem de programação Python, por ser uma linguagem simples e clara, atrai muitos programadores. Ela não fica somente com a simplicidade, pois por trás existe uma linguagem poderosa muito usada para administrar sistemas e desenvolver grandes projetos. O crescimento em seu uso veio por ser uma linguagem simples mais, objetiva, focando logo ao ponto, sem ter que dar muitas voltas em códigos para chegar a um resultado. Segundo a documentação oficial, "Python é uma linguagem de programação de propósito geral, de alto nível e que pode ser aplicada em muitos tipos diferentes de problemas" Python (2022). Essa afirmação faz jus à popularidade crescente da linguagem, usada para diversos fins,

dentre as quais: construção de sistemas Web com base em Frameworks, análise de dados, Machine Learning, inteligência artificial, entre outros.

A linguagem Python possui inúmeras vantagens, facilidade no aprendizado por possuir uma sintaxe muito acessível e sua criação ser baseada na agilidade. Nessa lógica passa a ser uma linguagem inicial fácil de se aprender, portátil e multiplataforma, a partir do seu interpretador é possível rodar em diversos sistemas operacionais; é gratuita, tem frameworks muito interativos, bem como comunidade com muitos exemplos para ajudar a iniciar.

Por outro lado, suas desvantagens encontram-se na velocidade interpretada, comparada a outras linguagens como C e C++ ela acaba ficando para trás, pois seu código precisa ser interpretado antes de ser executado, obtendo uma maior lentidão em determinadas aplicações. Outro fator que deixa a linguagem python em desvantagem, é o desenvolvimento móvel por não possuir uma biblioteca nativa para IOS e android.

### 2.4.1 Ambiente de execução

O ambiente de execução do projeto foi elaborado baseado em uma máquina com o sistema operacional Windows 10 19.03. Utilizou-se, ferramentas gratuitas para a criação e assim facilitar a busca de informações em fóruns, a documentação tanto das ferramentas como da linguagem escolhida foi encontrada em seus sites oficiais.

- A primeira etapa, criação de um ambiente virtual para utilizar a linguagem no mesmo.
- A Segunda etapa, seleção de ferramentas de codificação para trabalhar com o ambiente e a linguagem desejada.
- A terceira etapa, instalação dos pacotes de linguagem e o *framework* a ser utilizado.
- A quarta etapa, instalação das ferramentas de versionamento para controle de repositório e mudanças no sistema.
- A quinta etapa foi o desenvolvimento do produto.

Esse ambiente de execução foi elaborado após toda a diagramação e criação dos processos do software, assim ficando mais simples a implementação do código e os testes para validação dos requisitos iniciais. (SCUDERO, 2022)

## 2.5 Banco de dados

Conforme Date (2004, p.26) define banco de dados como um sistema computado-rizado para armazenar informações. De acordo com Heuser (2009, p.22), é um conjunto

de dados organizados para fins de gerenciamento e manipulação. Geralmente, os bancos de dados são usados para armazenar informações de vários tipos (como listas de clientes e produtos ou histórico de transações financeiras, por exemplo) bem como permitir que essas informações sejam acessadas de maneira adequada. Já o artigo de Scudeiro(2022), mostra que existem vários tipos de modelos de bancos de dados, tais como os orientados a documentos; orientados a chave/valor; orientados a colunas; orientados a grafos; e bancos de dados relacionais.

Para este projeto foi escolhido o banco de dados relacional SQLite por se tratar de um software de baixo tráfego de dados. Os bancos de dados relacionais usam tabelas para armazenar dados e cada uma pode conter vários campos que armazenam tipos específicos de informações, como o nome de um cliente, o número de um produto ou a data de uma transação. As tabelas são relacionadas entre si por meio de chaves, são campos que contêm valores únicos e podem ser usados para identificar uma linha em uma tabela.

De acordo com Machado (2020, p,39) e Heuser (2009, p.38), os bancos de dados relacionais permitem que as informações sejam facilmente recuperadas por operações de consulta, que são usadas para selecionar um conjunto de linhas de uma tabela e devem atender a certos critérios como, por exemplo, buscar todos os registros relacionados ao nome de determinado cliente. As consultas também podem ser usadas para agrupar e classificar os dados em uma tabela.

Segundo Machado (2020, p,40), os bancos de dados relacionais permitem que as informações sejam facilmente manipuladas por operações de inclusão, alteração e exclusão. Os bancos de dados relacionais são flexíveis e podem ser facilmente adaptados às mudanças nos dados armazenados. No entanto, os mesmos podem ser complexos de ser gerenciados e podem não ser adequados para algumas categorias específicas de aplicações.

## 2.6 Versionamento

Versionar um software é o processo de documentar as mudanças feitas durante o processo de criação do mesmo, na maioria das vezes, com base numérica. Esse processo de versionamento mostra as mudanças aplicadas no software para que os usuários finais entendam as melhorias e as novidades entregues a eles. Durante a execução deste projeto, foi utilizado o versionamento do Git, junto ao Github, ferramenta muito utilizada na área de desenvolvimento de softwares.

A ferramenta Git é um sistema *open-source* de controle de versão, podendo manter o histórico de mudanças nas ramificações criadas. O Git ajuda a controlar o fluxo de novas funcionalidades entre vários desenvolvedores, no mesmo projeto Fernandes (2022). O Github é um serviço online de hospedagem para esses códigos, mais comumente chamado de repositório. Esse método de trabalho promove enorme facilidade para os desenvolvedores.

Trabalha a sua ferramenta particular sem alterar a base do código principal. Além disso, ele indica a possível quebra de código na ramificação principal que é uma vantagem para prevenir possíveis falhas.

O Github possui algumas funcionalidades que chamam bastante atenção por exemplo, cópia de repositório entre contas de usuários, possibilitando disponibilizar para outros colaboradores interessados no projeto. Outra funcionalidade bastante usada é o *forking* que permite você bifurcar uma base já criada e mudar conforme o usuário desejar sem modificar o projeto original, dando vida a um trabalho novo. A ferramenta pode ser integrada a aplicações de terceiros criando um leque de possibilidades, como aplicativos de gerência de erros, sistemas de proteções de *branches*, sistema de notificações personalizadas. Além desses aplicativos (APP), dentro do Github, é possível integrar a outros sistemas voltados à organização e desenvolvimento de software, como Visual studio code, Atom, Slack e Microsoft team.

O sistema de *branches* do git, junto a interface web do Github, facilita a interação com o código. Logo, com ele é possível ver criar e excluir *branches*, clonar repositórios, escolher quais arquivos participam de cada *commit*, trazendo todo o histórico dos *commits*, apresentando-se como uma interface mais humana para facilitar as coisas.

E, por fim, a opção de *commits*. Trata-se de uma característica incrível que mostra as mudanças de arquivos com data e hora, possibilita a visualização não somente dos comentários mas também as alterações nas linhas dos códigos e deixa a opção de publicar um comentário em cima de cada commit para ajudar o desenvolvedor.

## 2.7 Trabalhos similares

Durante as pesquisas, os seguintes softwares para gerenciamento de lavagem de veículos foram encontrados.

### 2.7.1 Software proprietário

- Gestão Clic. Sistema proprietário com módulos de gestão, o software foi criado para atender várias demandas do segmentos de veículos, incluindo opções tais como, venda, financeiro, estoque e notas fiscais. Na parte de serviços, ele tem opções como, troca de óleo, controle de frota de viagem, manutenção em mecânicas entre outros. Trata-se de um software bem completo, porém complexo de se utilizar no dia a dia.
- Jump Car: O software proprietário que mais se aproximou do modelo do sistema desenvolvido foi a Jump Car. Ele possui módulos com o sistema de lavagens de veículos. Entretanto possui separadamente uma plataforma de gerência de contratos,

estoque entre as demais, comercializa isoladamente, aplicativo e marketing para o usuário.

### 2.7.2 Software código aberto

- mardy07/lavajato: O software lavajato é um sistema de código aberto. Ele está, atualmente, na plataforma do Github para implementação e possui funcionalidades que se assemelham com o projeto em desenvolvimento, mas, ainda faltam algumas básicas como cadastrar cliente e relatório financeiro e sua documentação é bem simples, com poucas informações precisas.
- milanezluca/sujinho: O software Sujinho Lava Car é um sistema de código aberto, e também está na plataforma Github para implementação. Suas funcionalidades não foram descritas na documentação e não possui muitos comentários em seu código, para melhor entendimento.
- Market Up: Em alguns estabelecimentos é utilizado o software Marketup. Ele é gratuito para pequenas e médias empresas por ser gratuito ele abrange um leque muito grande de funcionalidades fugindo da simplicidade, misturando desde venda de consumíveis a prestação de serviços.

## 3 Materiais e métodos

Este tópico tem por objetivo apresentar as principais ferramentas e tecnologias utilizadas no decorrer das etapas do projeto, do início até à entrega.

### 3.1 Materiais - Ferramentas e tecnologias utilizadas

#### 3.1.1 HEFLO

O Heflo é uma ferramenta online de documentação de processos de negócio, diagramas com a notação BPMN. Ao passo em que são editados os fluxos de processos online, documentos são gerados automaticamente, Hefflo (2022). Seu usuário possui planos gratuitos para uso acadêmico e pacotes de planos para os demais. Este projeto utilizou o plano acadêmico para elaborar os diagramas de ciclo de vida e processos presentes neste projeto.

#### 3.1.2 Visual Studio Code

O Visual Studio Code é um editor de código-fonte, disponível para Windows, macOS e Linux. Possui suporte interno para JavaScript e Node.js, além de um rico ecossistema de extensões. Microsoft (2022). O Visual Studio Code foi usado para escrever o código-fonte de todo o projeto.

#### 3.1.3 Git e Github

Git é um sistema de versionamento de arquivos com código aberto, efetivo para melhoria na velocidade de gerência de um projeto. O Github é uma plataforma de hospedagem de arquivos para versionamento do sistema. E possui funcionalidades como gerenciar e integrar projetos com outras ferramentas como Slack, Trello, Gitkraken. O Git é usado no controle de versões do projeto, ajudando no melhor gerenciamento das configurações e mudanças nos repositórios existentes.

#### 3.1.4 UML

UML ou *Unified Modeling Language* é uma linguagem usada para elaborar toda estrutura de determinados projetos. Possui um grande vocabulário visual junto a regras cuja finalidade é repassar a comunicação por meio de representações conceituais e físicas de um sistema. Voltada para visualizar, construir, especificar e documentar os componentes

de um sistema complexo de um projeto. A UML foi utilizada como modelagem padrão do sistema em apresentação.

### 3.1.5 Python

A linguagem de programação Python é atualmente uma das linguagens que mais crescem no mercado de programação, e isso é devido a sua biblioteca padrão ser bem abrangente e ter uma legibilidade agradável. A linguagem também é bem expressiva, sem ter uma imensa gama de códigos. Possui, ainda, bons frameworks para pequenos e grandes projetos.

Esse software utiliza o *framework* Django, com ele facilita a comunicação entre banco de dados e interface web, para diminuir a necessidade de codificação e entregar um projeto consistente de médio a grande porte.

### 3.1.6 Django

O Django é um *framework web full stack open source* (código aberto) baseado em Python, gratuito e de alto nível. Ele foi criado com o objetivo de resolver todos os problemas mais comuns do processo de desenvolvimento de aplicações web por exemplo, autenticação, rotas, *object relational mapper* (ORM) e até *migrations*. Além disso, ele ainda suporta projetos escaláveis e conta com um eficiente sistema de segurança.

### 3.1.7 Bootstrap

Bootstrap é um *framework frontend* responsável por fornecer a estrutura CSS para criação de sites e aplicações responsivas de forma simples. É utilizado tanto em sites para computadores desktops quanto para páginas de dispositivos móveis com mais facilidade e agilidade.

### 3.1.8 Persistência de dados com SQLite

O software de banco de dados simplifica o gerenciamento de dados, permitindo que os usuários armazenem eles em um formulário estruturado e depois os acessem. Ele normalmente tem uma interface gráfica para ajudar a criar e gerenciar os dados e, em alguns casos, os usuários podem construir os próprios bancos de dados usando o software do banco de dados. O SQLite é um banco de dados relacional, capaz de criar um banco de dados embutido, muito utilizado em sistemas mobile por ser bem leve.

## 3.2 Métodos

O processo de identificação do problema surgiu de conversas informais dentro de uma empresa local de lavagem de veículos, devido a constante reclamação de falta de um sistema de gerenciamento simples que atendesse a demanda de organização e informações da empresa, surgindo a primeira intenção de criar algo que suprisse tal necessidade de um software de controle. Como complemento de informações foi buscada pesquisa online, como forma de completar o entendimento da ideia inicial foi feita uma pesquisa local em outras empresas e nenhuma das pesquisas apontaram um sistema que pudesse preencher essa demanda.

Para completar foi utilizada a experiência profissional do autor que trabalhou no ramo de prestação de serviço na área de lavagem de veículos por cerca de 6 anos, e compreende as dificuldades abordadas, assim, aumentando a motivação de criação do software presente.

Após identificado o problema e limitações, foi possível levantar um escopo com as necessidades e requisitos que o projeto atenderia. Esse passo é importante por classificar as necessidades a serem resolvidas como uma lista, obtendo de maneira mais clara as especificações do projeto.

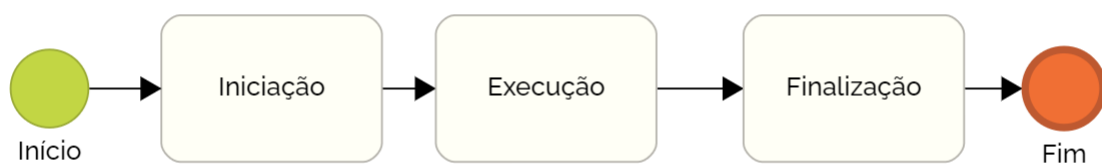
Todas especificações foram inseridas em um software para trabalhar a modelagem unificada, essa modelagem define quais os elementos, as entidades e o comportamento de toda a ferramenta a partir desse processo. Com os casos definidos, iniciou-se o desenvolvimento do software, colocando em prática a implementação das funcionalidades, visando alcançar o produto funcional.

Logo, a conclusão foi de um produto com as funcionalidades desejadas, passando por vários ajustes e implementações durante o processo de criação esse passo se relaciona diretamente ao objetivo geral proposto, assim podendo caminhar ao último objetivo que é implantar o mesmo em ambiente de produção.

### 3.2.1 Ciclo de vida e processos

O ciclo dos processos foi elaborado de modo a se ajustar à metodologia de desenvolvimento, dividindo-se em três etapas principais: iniciação, execução, finalização. Este software foi feito de forma simplificada, por possuir apenas um desenvolvedor sem necessitar usar métodos como *Daily Scrum* (Reunião Diária). Nesse sentido as *Sprints* não foram datadas ao final de cada *commit* foi elaborada uma pequena retrospectiva gerando as revisões.

Figura 1 – Processos

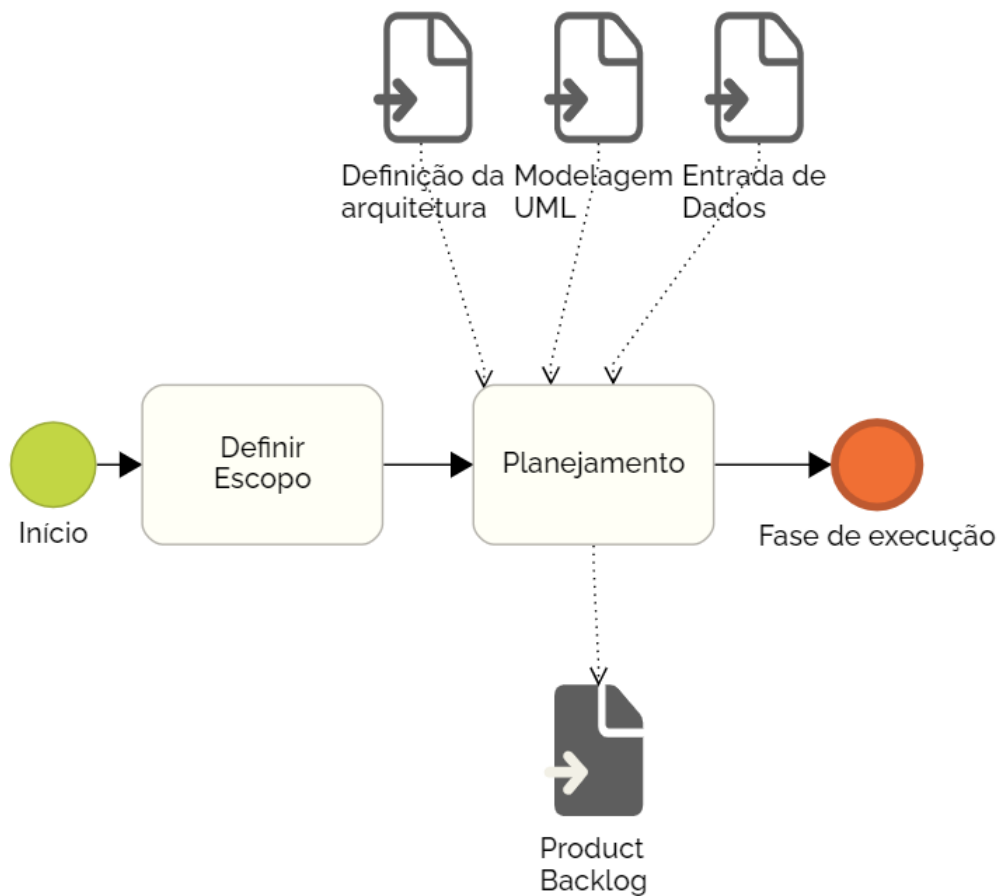


Fonte: Elaborado pelo autor, 2022

### 3.2.2 Fase de iniciação

Na fase de iniciação define-se o escopo da arquitetura e modelagem UML figura 2.

Figura 2 – Fase de iniciação

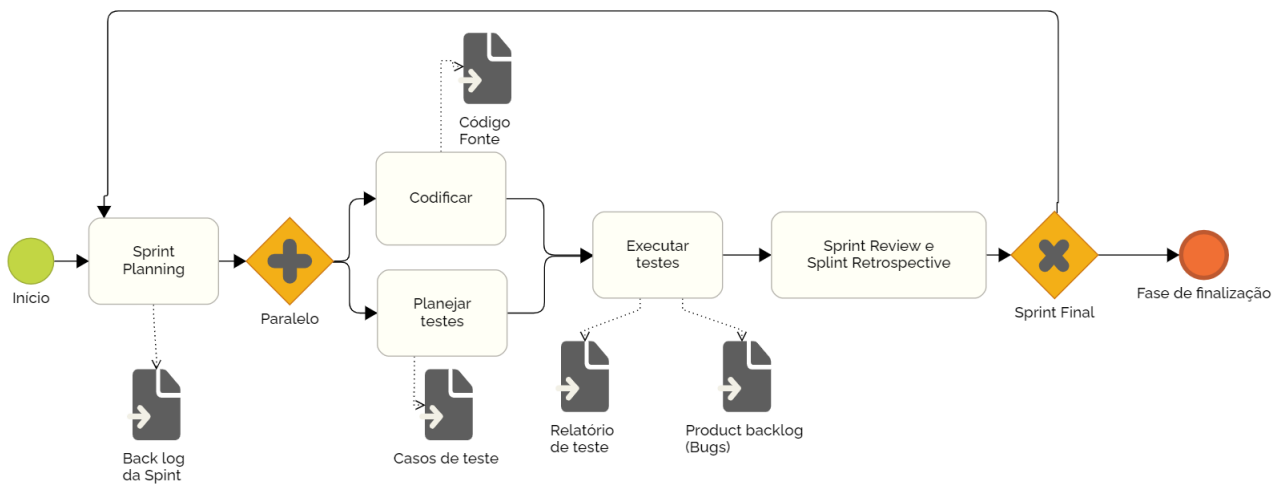


Fonte: Elaborado pelo autor, 2022

### 3.2.3 Fase de execução

A fase de execução figura 3, define os requisitos desenvolvidos esses requisitos resultaram em código-fonte, levando a processos de testes em produção assim entrando para lista de possíveis revisões ao final da sprint, o processo se repete até finalizar o ciclo de execução.

Figura 3 – Fase de execução

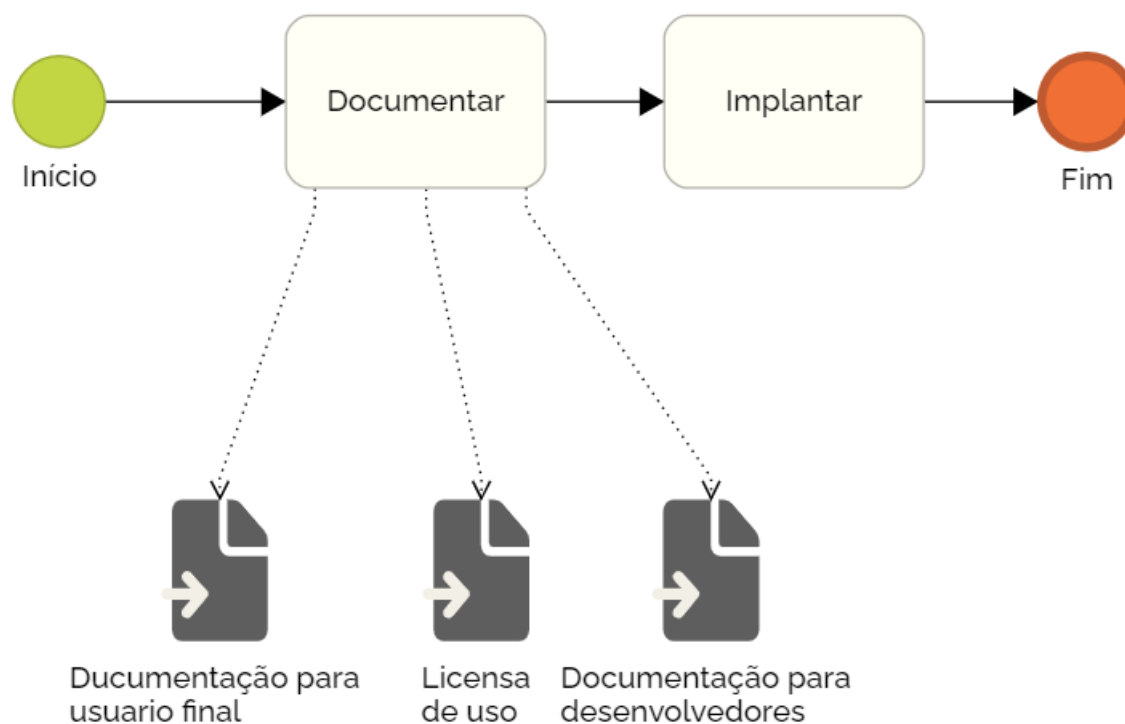


Fonte: Elaborado pelo autor, 2022

### 3.2.4 Fase de finalização

O último ciclo, finalização figura 4, é a criação de documentos voltados ao usuário final e desenvolvedores. Nessa etapa, cria-se a licença de uso de software. Após esses passos todo o desenvolvimento foi finalizado e o software finalizado para disponibilizar ao cliente final.

Figura 4 – Fase de finalização



Fonte: Elaborado pelo autor, 2022

### 3.2.5 Controle de Versão

O controle de versão foi feito por meio do Github, uma aplicação global de controle de código-fonte com base no Git. A partir dessa ferramenta foi possível desenvolver novas funcionalidades, criando *branches* isoladas e implementando, sem a preocupação de quebrar a *branch* principal. Após o término da implementação de cada funcionalidade é feita uma *pull request* dessa *branch* para a *branch* principal. Com a *pull request* é possível comparar todas as alterações feitas, validar e por fim, aprovar. Após a aprovação a *pull request* é fechada e o código é mesclado à *branch* principal.

### 3.2.6 Modelagem

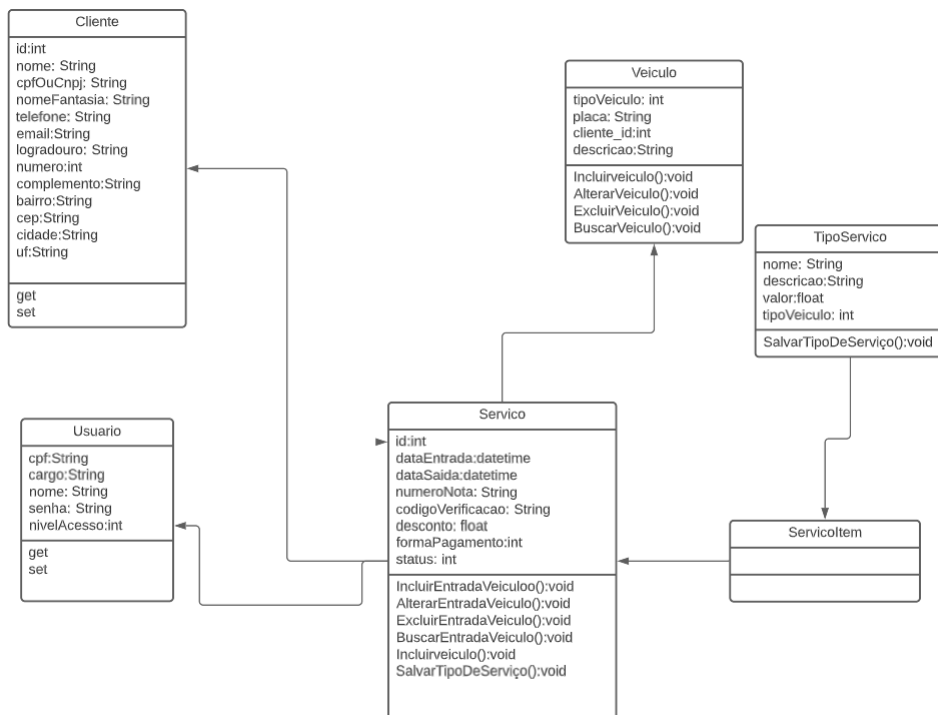
Modelagem de dados segundo Sap (2022) “é o processo de diagramação de fluxo de dados”. Ao iniciar uma nova instância de banco de dados o responsável pelo projeto elabora um diagrama com o intuito de mostrar o fluxo de informações de entrada e saída que vai

ocorrer no banco. Assim, com essas informações o fluxo determina as características e os formatos dos dados, suas estruturas e funções de processamento, visando atender os requisitos com maior eficiência. Após pronto o modelo de dados criado se torna parte da documentação do sistema e de maneira simples, mostra como funciona todo o processo de backend das informações.

O modelo de dados ajuda a criar o banco de dados lógicos e remove possíveis redundâncias, reduz os requisitos de armazenamento e permite uma melhor recuperação de dados perdidos. Antes de construir ou projetar qualquer software novo, deve-se criar uma visão documentada de como o software vai se comportar e de como é a sua visão de produto final. Essa parte são as regras de negócios junto as funcionalidades levantadas. Em seguida, é feita a descrição do fluxo de dados e de como o banco de dados vai suportá-lo. A modelagem de dados registra definitivamente a visão para o projetista, a partir dessa visão o software pode ser desenvolvido conforme as especificações e deve fornecer a nitidez esperada nos dados.

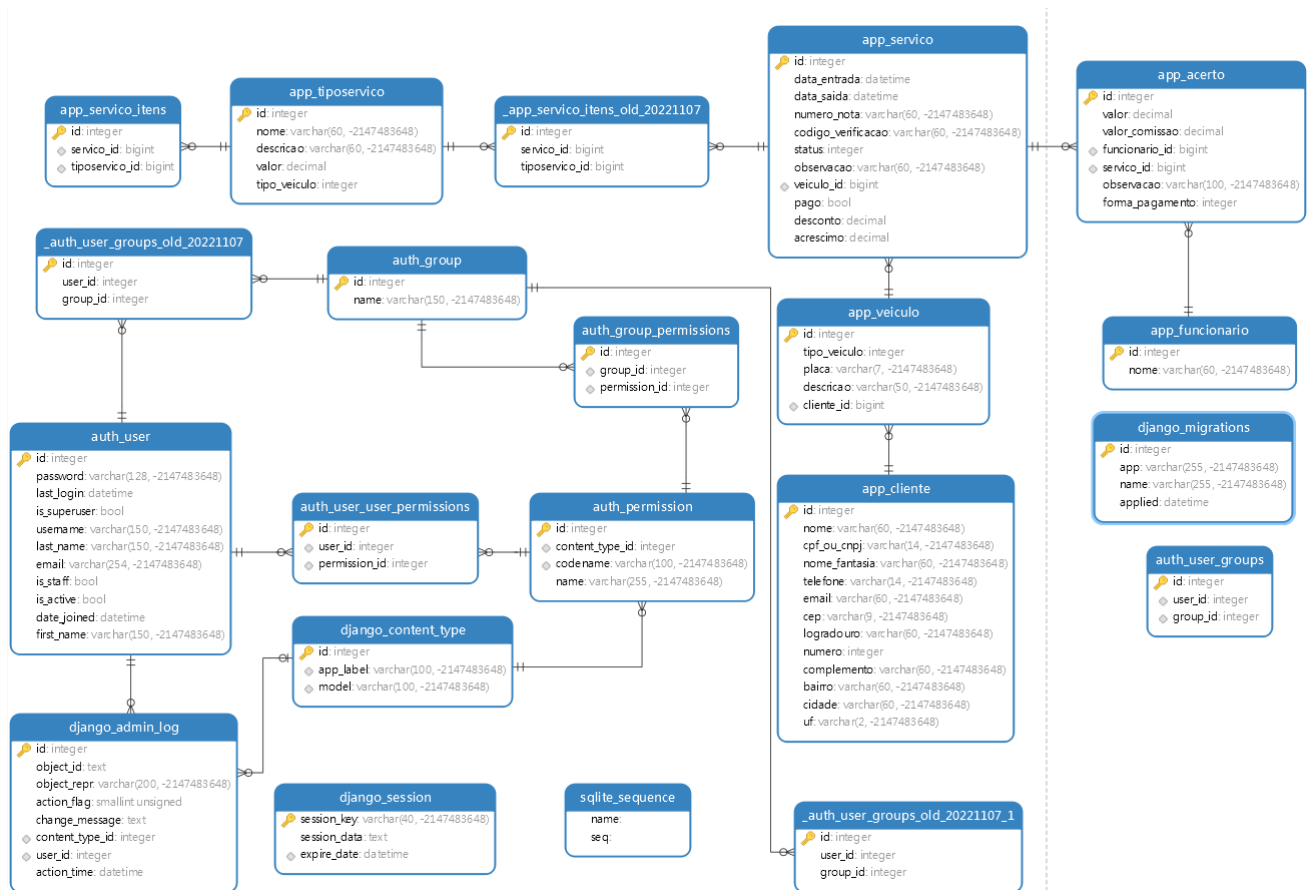
Durante todo o processo de documentação do software, diagramas UML foram criados dentre eles o diagrama de classes responsável por modelar a estrutura e relacionamento das classes que foram utilizadas como modelo para a criação dos objetos utilizados na aplicação.

Figura 5 – Diagrama de classe



Modelo de banco.

Figura 6 – Modelagem de dados



Fonte: Elaborado pelo autor, 2022

### 3.3 Requisitos

Requisitos de software é o estudo de viabilidade do software, é levantado e verificado todas as exigências do software, e como se encaixam seus objetivos, como serão alocados seus recursos. Tudo isso com informações levantadas da necessidade do problema proposto a ser resolvido. Satisfazendo essas expectativas, a confiabilidade do software terá uma garantia maior.

#### 3.3.1 Requisitos funcionais

- O usuário deve ser capaz de fazer login com um usuário existente criado pela gerência.

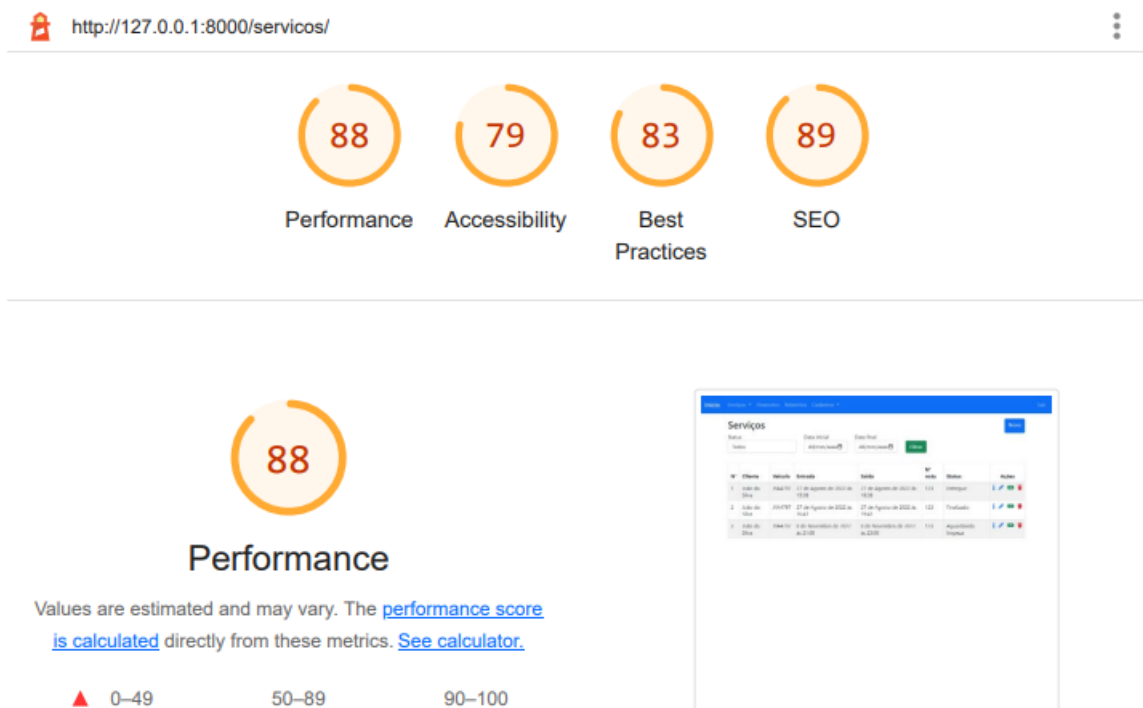
- O usuário deve ter acesso a uma tela inicial com os serviços abertos dos últimos 3 dias.
- O usuário deve ter acesso a um cadastro de tipos de serviço, permitindo criar, editar e remover.
- O usuário deve ter acesso a um cadastro de clientes, permitindo criar, editar e remover.
- O usuário deve ter acesso a um cadastro de veículos, permitindo vincular mais de um veículo a um cliente e também criar, editar e remover.
- O usuário deve ter acesso a uma lista de serviços com filtros de status, data inicial e data final.
- O usuário deve ter acesso a um lançamento de serviço o qual permite criar e editar conforme as etapas da lavagem ocorram.
- O usuário deve ter acesso a um resumo financeiro de cada serviço.
- O usuário deve ter acesso a uma tela que exiba apenas os serviços em andamento.
- O usuário deve ter acesso a um formulário de acerto financeiro para informar dados do pagamento e de comissão.
- O usuário deve ser capaz de fazer mais de um acerto para um serviço, permitindo várias formas de pagamento.
- O usuário deve ser capaz de redefinir todos os lançamentos financeiros para que sejam refeitos.
- O usuário deve ter acesso a exclusão de um serviço e todos os acertos do mesmo de maneira recursiva.
- O usuário deve ter acesso a uma tela de lançamentos financeiros a qual categorize por cores os registros em andamento e finalizados.

### 3.3.2 Requisitos não funcionais

- O sistema deve ser executado na plataforma web.
- O sistema deve ser responsivo de forma que possa ser utilizado por computadores, notebooks, tablets e smartphones.
- O sistema deve ser capaz de funcionar em todas os navegadores, exceto Internet Explorer.

- O sistema deve ter notas acima de 80 em todos os testes Lighthouse.

Figura 7 – Testes Lighthouse



Fonte: Elaborado pelo autor, 2022

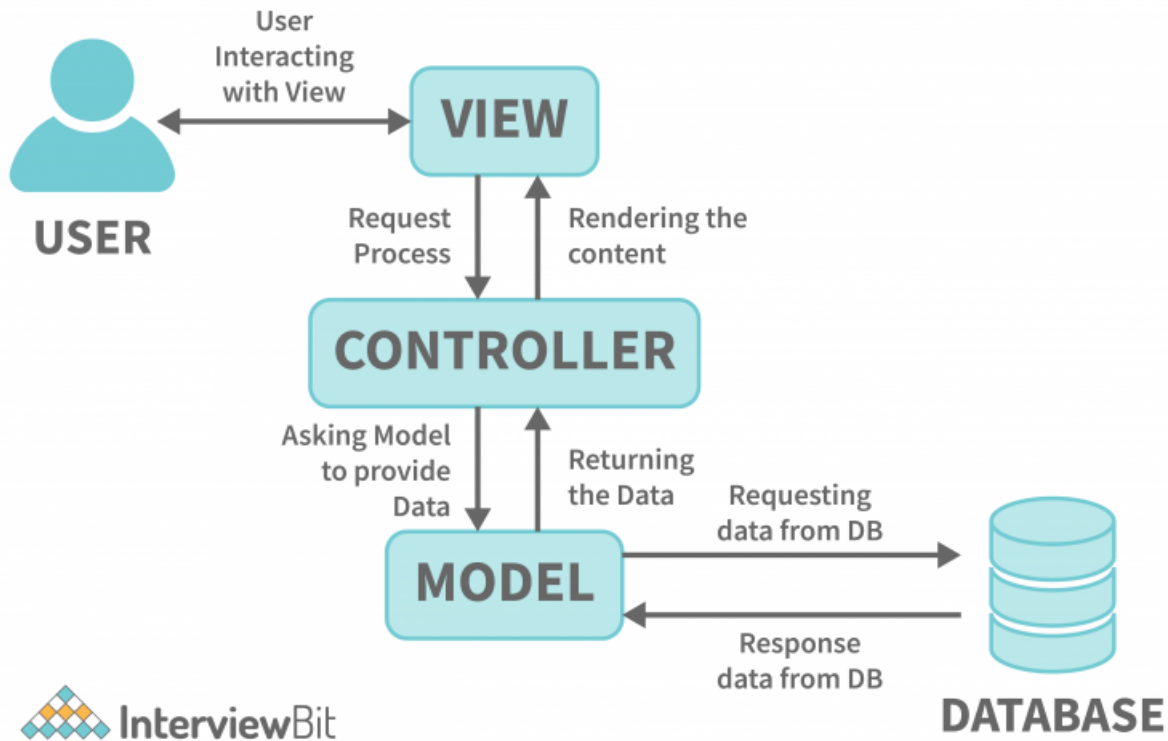
O Lighthouse é uma ferramenta de código aberto que faz a leitura do código e audita automaticamente mostrando o necessário para melhorar a página web. Como análises de desempenho, acessibilidade e compatibilidade.

### 3.4 Arquitetura do software

MVC, conhecida como, *Model-View-Controller* (modelo-visão-controle), é um padrão de projeto de software, muito popular na construção de aplicativos. Esse padrão divide o aplicativo em seções menores. Apesar de ser um bom modelo, também possui desvantagens:

- Em sistemas de baixa complexidade o produto viável mínimo (MVP) pode aumentar a dificuldade se tornando desnecessária.
- No momento que é selecionada é necessário sempre estar atento à separação de camadas para não misturar,
- E por fim, o tempo de modelagem pode requerer um tempo maior.

Figura 8 – Arquitetura MVC



Fonte: (INTERVIEWBIT, 2022)

O projeto foi desenvolvido utilizando o *framework* Django. Por padrão esta tecnologia utiliza a arquitetura MVC para a divisão de responsabilidades em cada camada do projeto. A sigla MVC significa *Model View Controller* e tem como princípios as seguintes responsabilidades:

- **Model:** Criar modelagem objeto relacional e criar o acesso aos dados tanto para escrita quanto leitura;
- **View:** Renderizar as visualizações da aplicação ao usuário e captar todas as interações, requisições e respostas vindas do controller;
- **Controller:** Receber a requisição do usuário, aplicar as regras de validação e repassar a requisição à model. Por fim o controller redireciona a uma view ou devolve dados serializados.

## 3.5 Licença de uso

A licença escolhida foi a MIT<sup>1</sup>, criada pelo Instituto de Tecnologia de Massachusetts (MIT). Ela é muito utilizada, tendo a possibilidade de ser tanto para software livre como para proprietário.

Atualmente é a licença mais popular no Github por ser expressivamente curta e fácil de entender, com ela é permitido que qualquer pessoa faça qualquer mudança desde que mantenha uma cópia da licença incluindo direitos autorais e os avisos a escolha foi baseada na possibilidade de o software ser escolhido como base dependentes de outros possíveis projetos.

---

<sup>1</sup> anexo A



## 4 Resultados e discussões

### 4.1 Gerenciamento de configuração e mudanças

O gerenciamento de configuração e mudanças do projeto em toda a parte de programação foi feito por meio do Git e Github. Para realizar alterações no projeto foram criadas *branches* com prefixos feat/NomeMudança e quando prontas, foram mescladas com a *branch* main por meio de *pull requests*. O gerenciamento de mudanças, voltado para parte estrutural, foi feito ao final de cada modificação significativa nas demandas. Cada funcionalidade ou requisitos elaborados foram validados considerando alguns fatores:

- **Processos** - Identifica a possibilidade de ficar mais oneroso a mudança.
- **Estrutura organizacional** - Identifica a possibilidade de ficar fora dos requisitos iniciais;
- **Funções de trabalho** - Identifica se é encaixado dentro do que a empresa atua;
- **Comportamentos** - Como será o comportamento dos objetos e recursos envolvidos.

### 4.2 Processo de desenvolvimento

Reis (2001, p3) afirma que o processo de um software é uma sequência coerente de tarefas ou fases para a construção de um software. As fases que compõem um processo de software são:

- **Requisitos:** elicitam as necessidades ou requisitos operacionais e descrevem as funções desejadas no sistema a ser executado.
- **Projeto de sistema:** tradução dos requisitos em forma de descrição e todos os componentes necessários para a implementação do código do sistema.
- **Implementação do código:** produção do código que controla o sistema e faz a computação lógica envolvida.
- **Verificação das funções:** valida em forma de *Checkout* todas as funções levantadas se correspondem com o que está sendo produzido.

- **Manutenção:** o software começa a gerar logs de melhorias repetindo os ciclos anteriores.

Com esses Passos foi possível criar um software de entrada para visualização esses ciclos se repetem até conseguir um resultado final significativo.

## 4.3 Documentação

O objetivo deste tópico é orientar e informar tanto o usuário final quanto o desenvolvedor sobre a utilização do sistema.

### 4.3.1 Documentação para usuários

O sistema está em um repositório do GitHub, por meio da opção wiki<sup>1</sup> dentro da plataforma GitHub. Foi criada toda a documentação de usuário, explicando o fluxo de cadastros e de geração de serviços. Está disponível através do link no rodapé. Essa documentação serve para o bom uso e entendimento do software.

### 4.3.2 Documentação para desenvolvedores

A documentação de todo o processo de implementação para a aplicação será descrita abaixo com objetivo de instalar o software do zero sem dados em seu banco.

1. Clonar ou baixar repositório do Github no link do rodapé.
2. Instalar o Python 3.8 ou superior.

**Windows:** abrir o site oficial do Python ir até a aba *downloads* baixar o cliente de instalação, iniciar a instalação habilitar a opção Add python.exe to PATH, essa opção implementa algumas facilidades no encaminhamento de nomes, aguardar carregar todo o python e clicar no botão close.

**Linux:** abrir o terminal e inserir o comando `apt-get install python3.8`, esperar carregar a aplicação e, em seguida, utilizar o comando `python3 --version` para validar a versão instalada.

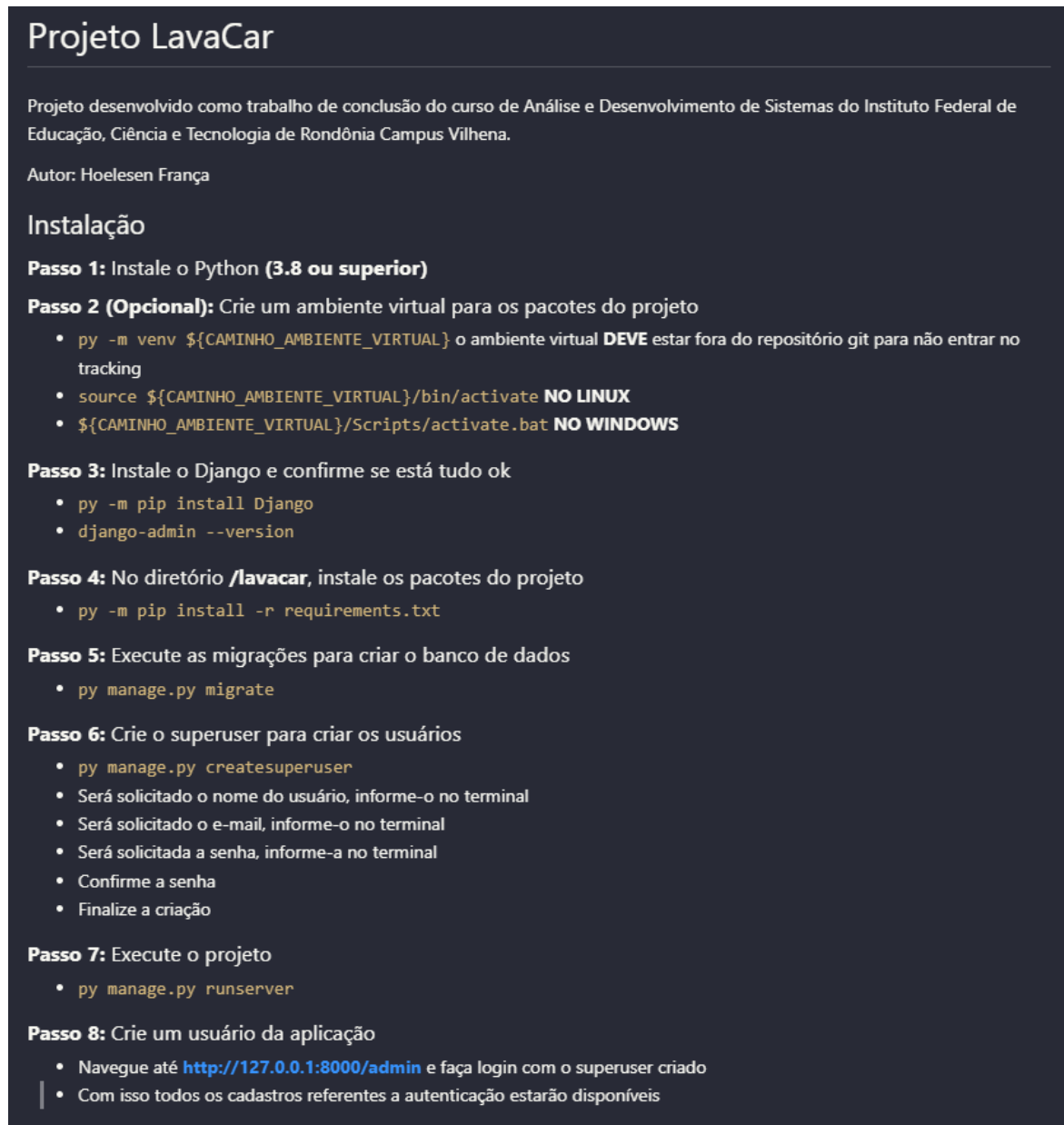
3. Opcional, crie um ambiente virtual para os pacotes do projeto.
  - `py -m venv CAMINHO-AMBIENTE-VIRTUAL` - O ambiente virtual deve estar fora do repositório git para não entrar no tracking.
  - `source /CAMINHO-AMBIENTE-VIRTUAL/bin/activate` - Ambiente Linux.
  - `CAMINHO-AMBIENTE-VIRTUAL/Scripts/activate.bat` - Ambiente Windows.

<sup>1</sup> <https://github.com/Hoelesen/LavaCar/wiki/0—Bem-vindo-a-Lava-Car>

4. Instale o Django e confirme se está tudo ok.
  - `py -m pip install Django` .
  - `django-admin --version` .
5. No diretório /lavacar, instale os pacotes do projeto.
  - `py -m pip install -r requirements.txt` .
6. Executar as migrações para criar o banco de dados.
  - `py manage.py migrate` .
7. Criar o super usuário ele é o responsável por criar os demais usuarios da aplicação..
  - `py manage.py createsuperuser` .
  - será solicitado o nome do usuário, informe-o no terminal.
  - Será solicitado o e-mail, informe-o no terminal.
  - Será solicitada a senha, informe-a no terminal.
  - Confirme a senha.
  - Finalize a criação.
8. Execute o projeto.
  - `py manage.py runserver` .
9. Crie um usuário da aplicação.
  - Navegue até `http://127.0.0.1:8000/admin` e faça login com o super usuário criado.
  - Com isso todos os cadastros referentes a autenticação estarão disponíveis.

Este é o processo para subir a aplicação, esse documento também está disponível no repositório Github no arquivo README.md.

Figura 9 – Arquivo README.md



## Projeto LavaCar

Projeto desenvolvido como trabalho de conclusão do curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia Campus Vilhena.

Autor: Hoeslen França

### Instalação

**Passo 1:** Instale o Python (**3.8 ou superior**)

**Passo 2 (Opcional):** Crie um ambiente virtual para os pacotes do projeto

- `py -m venv ${CAMINHO_AMBIENTE_VIRTUAL}` o ambiente virtual **DEVE** estar fora do repositório git para não entrar no tracking
- `source ${CAMINHO_AMBIENTE_VIRTUAL}/bin/activate` **NO LINUX**
- `${CAMINHO_AMBIENTE_VIRTUAL}/Scripts/activate.bat` **NO WINDOWS**

**Passo 3:** Instale o Django e confirme se está tudo ok

- `py -m pip install Django`
- `django-admin --version`

**Passo 4:** No diretório `/lavacar`, instale os pacotes do projeto

- `py -m pip install -r requirements.txt`

**Passo 5:** Execute as migrações para criar o banco de dados

- `py manage.py migrate`

**Passo 6:** Crie o superuser para criar os usuários

- `py manage.py createsuperuser`
- Será solicitado o nome do usuário, informe-o no terminal
- Será solicitado o e-mail, informe-o no terminal
- Será solicitada a senha, informe-a no terminal
- Confirme a senha
- Finalize a criação

**Passo 7:** Execute o projeto

- `py manage.py runserver`

**Passo 8:** Crie um usuário da aplicação

- Navegue até <http://127.0.0.1:8000/admin> e faça login com o superuser criado
- Com isso todos os cadastros referentes a autenticação estarão disponíveis

Fonte: Elaborado pelo autor, 2022

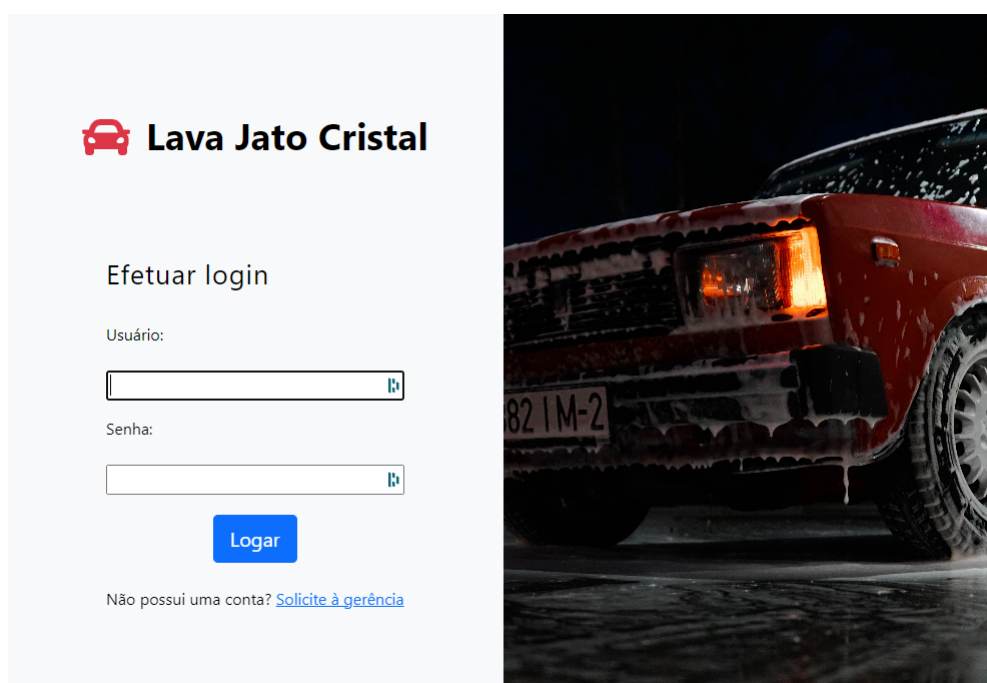
## 4.4 Demonstração do software

Neste capítulo é demonstrado as telas do software com uma pequena descrição cada e suas funcionalidades para melhor entender o fluxo do sistema.

### 4.4.1 Login

Ao criar o projeto também é criado o superusuário para dar acesso acesso à área de administração do Django, para assim criar os usuários que vão logar no sistema e os funcionários que irão aparecer na lista de serviços.

Figura 10 – Tela de login



Fonte: Elaborado pelo autor, 2022

Após efetuado login é possível ver uma tela inicial com as informações de navegação da aplicação web.

Figura 11 – Tela inicial

**Seja bem vindo**  
Este é o portal de serviço, um portal onde você pode acessar os recursos disponíveis

NOVO SERVIÇO

SERVIÇOS EM ANDAMENTO

ENTRADAS E SAÍDAS DIÁRIAS

AGENDAMENTOS A RECEBER

RELATÓRIOS

CADASTROS

Digite sua busca

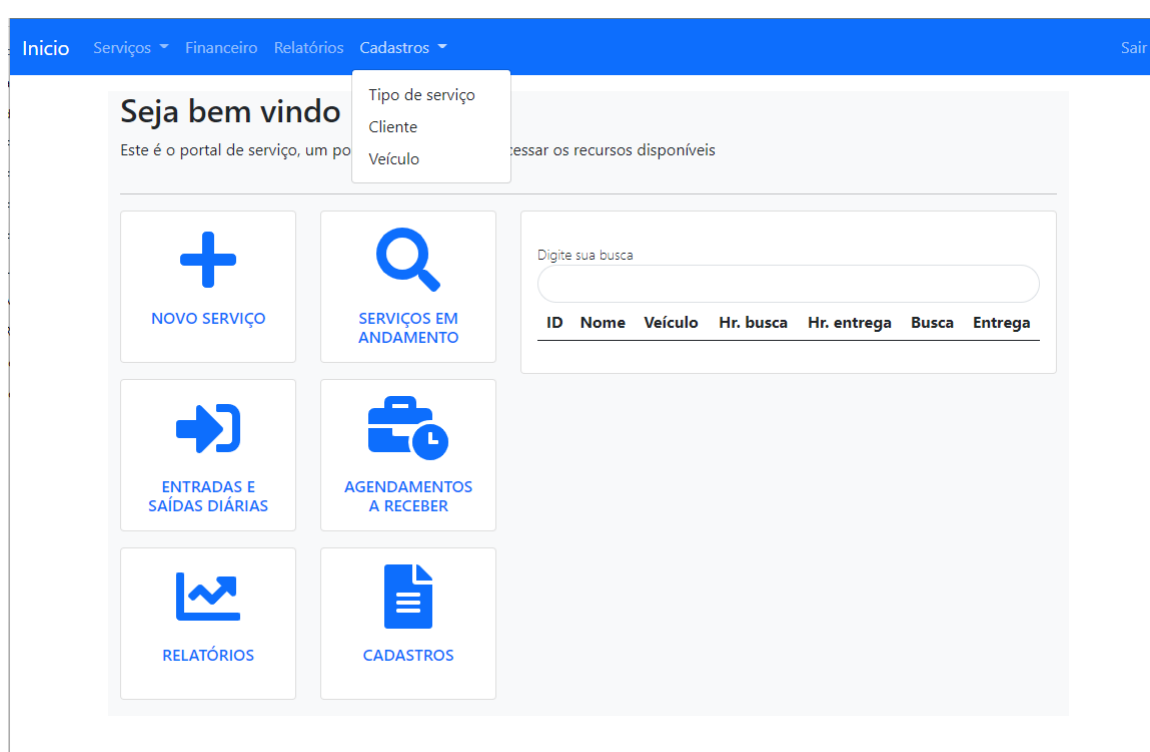
ID	Nome	Veículo	Hr. busca	Hr. entrega	Busca	Entrega
4	João pedro da Silva	KHH1354	15/11 09:26	16/11 12:00	<input type="checkbox"/>	<input type="checkbox"/>
5	Hoelesen França	CLM8525	17/11 10:30	17/11 11:30	<input type="checkbox"/>	<input type="checkbox"/>
6	João pedro da Silva	KHH1354	15/11 08:30	15/11 09:30	<input type="checkbox"/>	<input type="checkbox"/>
7	Pedro Santiago	MJJ-000	15/11 09:34	15/11 10:34	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: Elaborado pelo autor, 2022

## 4.4.2 Cadastros

A tela Inicial tem a opção cadastros e três submenus, por qual se inserem as informações necessárias para geração de um novo serviço.

Figura 12 – Menu cadastros



Fonte: Elaborado pelo autor, 2022

Nessa tela é possível cadastrar novos clientes e visualizar os clientes já cadastrados. Na opção de cadastro.

Figura 13 – Cadastro de cliente

CPF/CNPJ	Nome	Nome fantasia	Telefone	Ações
00000000000	Pedro Santiago	Confecções pedro	69 9 0000-0001	
85974765249	Hoelesen França	Japa	00000000	
00000000000	João pedro da Silva	Pedro Industria	00 0 0000-0000	

Fonte: Elaborado pelo autor, 2022

Nessa tela é possível cadastrar novos veículos e visualizar os já cadastrados.

Figura 14 – Cadastro de veículo

Placa	Cliente	Tipo	Descrição	Ações
JYA5797	Japa	2	Fox Branco	
JJJ-000	Confecções pedro	1	Titan 150 branca	
HHH1354	Pedro Industria	3	Ranger Azul	
KLM8525	Japa	2	Gol bola preto	

Fonte: Elaborado pelo autor, 2022

Tipo de serviço é o que vai ser prestado ao cliente, lavagem, limpeza, polimento entre outros, esses serviços são criados já com seu valor estipulado e o tipo de veículo vinculado.

Figura 15 – Tipo de serviço



Nome	Valor	Tipo de veículo	Ações
Lavagem moto	30,00	1	 
Lavagem carro	50,00	2	 
Lavagem e Limpeza Caminhonete	90,00	3	 
Lavagem e limpeza Veiculo passeio	50,00	2	 
Lavagem caminhonete	50,00	3	 

Fonte: Elaborado pelo autor, 2022

A opção de tipo de serviço não se limita somente à parte de veículos empresas que prestam serviços de lavagem têm por costume também ofertarem serviços como desmontagem de carros, lavagens de tapetes, polimentos de pinturas, com a opção de cadastrar novos tipos de serviço o usuário poderá personalizar o que deseja.

### 4.4.3 Serviços

No menu serviço, adicionar serviço cadastra-se o serviço do cliente com as informações anteriormente já coletadas, inserindo o veículo do cliente, a data de entrada e data prevista para sair o veículo, o status classificados com a fase do serviço, exemplo se um serviço acabou de dar entrada o mesmo está com o status aguardando lavagem, caso optar somente por limpeza o status deve ser iniciado com a opção aguardando limpeza. Os valores criados em tipo de serviço estão na opção itens para seleção e também é possível inserir um desconto ou acréscimo prévio caso o serviço tenha alguma observação que gere custo a empresa.

Figura 16 – Adicionar serviço

Veículo  
Selecione

Data entrada  
dd/mm/aaaa --:--

Data saída  
dd/mm/aaaa --:--

N° da nota  
N° da nota

Código de verificação  
Código de verificação

Status  
AGUARDANDO\_LAVAGEM

Observação  
Observação

Itens  
 Lavagem moto - 30.00  
 Lavagem carro - 50.00  
 Lavagem e Limpeza Caminhonete - 90.00  
 Lavagem e limpeza Veiculo passeio - 50.00  
 Lavagem caminhonete - 50.00  
 Lavagem Caminhao - 90.00

Pago

Desconto  
0

Acréscimo  
0

Voltar Salvar

Fonte: Elaborado pelo autor, 2022

Seguindo em serviço, a opção todos os serviços é possível visualizar o serviço cadastrado nele temos quatro ícones ao lado esquerdo na coluna ações, o ícone com o

símbolo de exclamação contém um relatório de alterações no pagamento e funcionários que fizeram o serviço, o ícone do lápis é a opção de editar as informações inseridas ao criar o serviço, o ícone com o símbolo do dinheiro é a opção que se acerta os valores de pagamentos já efetuados e pendentes do serviço, e o último ícone com o símbolo da lixeira é onde se é possível apagar o serviço já criado.

Figura 17 – Todos serviços

N°	Cliente	Veículo	Entrada	Saída	N° nota	Status	Ações
2	Hoelesen França	CYA5797	13 de Outubro de 2022 às 11:23	13 de Outubro de 2022 às 14:23		Finalizado	
3	Pedro Santiago	MJJ-000	13 de Outubro de 2022 às 15:25	13 de Outubro de 2022 às 16:00		Lavagem em andamento	
4	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 09:26	16 de Novembro de 2022 às 12:00		Aguardando lavagem	
5	Hoelesen França	CLM8525	17 de Novembro de 2022 às 10:30	17 de Novembro de 2022 às 11:30		Aguardando lavagem	
6	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 08:30	15 de Novembro de 2022 às 09:30		Finalizado	
7	Pedro Santiago	MJJ-000	15 de Novembro de 2022 às 09:34	15 de Novembro de 2022 às 10:34		Finalizado	

Fonte: Elaborado pelo autor, 2022

#### 4.4.4 Financeiro

E, por fim, temos a tela do financeiro podendo filtrar os serviços já pagos e os pendentes, conforme alterações na tela anterior na qual era possível editar o valor do serviço impacta totalmente no filtro efetuado nesta tela, caso falte algum valor do serviço, a tela mostrará o mesmo como pendente. Como tem a possibilidade de filtrar por status e por data.

Figura 18 – Financeiro

### 4 Financeiro

Hoelesen França edited this page in 2 minutes · 2 revisions

E por fim temos a tela do financeiro onde é possível filtrar os serviços já pagos e os pendentes, conforme alterações na tela anterior onde se era possível editar o valor do serviço impacta totalmente no filtro efetuado, nesta tela caso falte algum valor do serviço essa tela mostrará o mesmo como pendente. Nessa tela é possível filtrar por status e por data.

Início
Serviços ▾ Financeiro Relatórios Cadastros ▾
Sair

#### Financeiro

Status

Data inicial

Data final

Filtrar

N°	Cliente	Veiculo	Entrada	Saída	N° nota	Status	Valor pendente	Valor acertado
2	Hoelesen França	CYA5797	13 de Outubro de 2022 às 11:23	13 de Outubro de 2022 às 14:23		Pago	R\$ 0,00	R\$ 50,00
3	Pedro Santiago	MJJ-000	13 de Outubro de 2022 às 15:25	13 de Outubro de 2022 às 16:00		Pago	R\$ 0,00	R\$ 30,00
4	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 09:26	16 de Novembro de 2022 às 12:00		Pendente	R\$ 108,00	R\$ 0
5	Hoelesen França	CLM8525	17 de Novembro de 2022 às 10:30	17 de Novembro de 2022 às 11:30		Pendente	R\$ 45,00	R\$ 0
6	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 08:30	15 de Novembro de 2022 às 09:30		Pendente	R\$ 90,00	R\$ 0
7	Pedro Santiago	MJJ-000	15 de Novembro de 2022 às 09:34	15 de Novembro de 2022 às 10:34		Pago	R\$ 0,00	R\$ 30,00

É isso tenha um bom uso tanto para implantação como para uso do projeto como base para novos projetos.

Essa wiki é a documentação para uso do projeto Lava\_car, projeto desenvolvido como tese de final de curso e solução para empresas locais, para acessar ao lado direito tem os guia por páginas.

Fonte: Elaborado pelo autor, 2022

## 5 Considerações finais

Como pôde ser observado no decorrer deste processo, pode-se entender que a criação do sistema de controle de lavagem de veículos foi alcançada, o software conseguiu chegar a um patamar de um produto mínimo que atende algumas necessidades e dificuldades das empresas de lavagem de veículos. As ferramentas adequadas facilitaram o desenvolvimento do sistema web, considerando a modelagem com vários elementos. Ferramentas como Bootstrap e Django foram decisivas para um desenvolvimento mais fluido e o banco de dados em Sqlite sendo leve e prático de se usar. O trabalho em conjunto de da linguagem com as ferramentas selecionadas tornou possível a criação do sistema web, o qual se mantém em um padrão visual com a característica de responsividade, de tal forma que o usuário possa utilizar em diversos dispositivos desde que tenha um navegador web.

### 5.1 Trabalhos futuros

O software chegou a um produto viável para implementação. Todavia ainda é possível notar algumas melhorias futuras, abaixo a lista de algumas das melhorias já levantadas.

- Recuperação de senha por email.
- Definir permissões de usuários pela aplicação e não pela tela administrativa do Django.
- Alertas de prazos de entrega de veículos.
- Interface de agendamento de busca e entrega.
- Criar API para comunicação com outros software.
- Criar módulo gestor de estoque.
- Habilitar modulo de relatórios com filtros melhores.
- Criar repositório de imagens de produtos.
- Gerar relatório formato csv para download.
- Disponibilizar o software para clientes do estabelecimento.
- Criar filtros para cadastros já efetuados.



# Referências

- BECKMAN, T. The current state of knowledge. in. liebowitz, j. *Knowledge management handbook*. New York: CRC, 1999.
- CÔRTEZ, M. L.; CHIOSSI, T. C. *Modelos de qualidade de software*. [S.l.]: Ed. da Unicamp, 2001.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier Brasil, 2004.
- FERNANDES, D. *Git Github: O que é? Por que? Como iniciar?* 2022. Url <https://blog.rocketseat.com.br/iniciando-com-git-github/>. Acesso em: 10/09/2022.
- HEFLO. *Documentação de processos de negócio*. 2022. Url <https://www.heflo.com/pt-br/ferramenta-de-documentacao-de-processos-de-negocio/>. Acesso em: 02/010/2022.
- HEUSER, C. A. *Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS*. [S.l.]: Bookman Editora, 2009.
- INTERVIEWBIT. *MVC Architecture – Detailed Explanation*. 2022. Url <https://www.interviewbit.com/blog/mvc-architecture/>. Acesso em: 08/11/2022.
- MACHADO, F. N. R. *Banco de Dados–Projeto e Implementação*. [S.l.]: Saraiva Educação SA, 2020.
- MICROSOFT. *Suas ideias merecem as melhores ferramentas*. 2022. Url <https://visualstudio.microsoft.com/pt-br/vs/>. Acesso em: 08/11/2022.
- ORG, P. *Python Geral*. 2022. Url <https://docs.python.org/pt-br/dev/faq/general.html>. Acesso em: 12/07/2022.
- REIS, C. Caracterização de um modelo de processo para projetos de software livre. *Mestrado (Dissertação)*. ICMS, Universidade de São Paulo, 2001.
- SAP. *O que é modelagem de dados?* 2022. Url <https://www.sap.com/brazil/insights/what-is-data-modeling.html>. Acesso em: 08/11/2022.
- SCUDERO, E. *TOP 10 principais SGBDs do mercado global!* 2022. Url <https://becode.com.br/principais-sgbds/>. Acesso em: 13/10/2022.
- SOMMERVILLE, I. *Engenharia de Software. Oitava Edição*. [S.l.]: São Paulo, Pearson Addison Wesley, 2007.



# Apêndices



# APÊNDICE A – Documentação para usuários

Nessa documentação mostrará as informações necessárias, para utilização do software aqui desenvolvido. Nas imagens abaixo mostra parte por parte para criação de um novo serviço de forma simples. Essa documentação esta no repositório do software, na opção de wiki interna do Github.

1- Tela de login do Software.

Figura 19 – Tela de login software

## 0 Bem vindo a Lava Car

Hoelesen França edited this page 6 days ago · 2 revisions

Edit New page

Bem vindo ao projeto Lava Car onde será encaminhado as informações necessárias para utilização do software aqui desenvolvido. Nas páginas abaixo mostrarei parte por parte para criação de um novo serviço de forma simples.



Pages 5

Find a page...

- 0 Bem vindo a Lava Car
- 1 Login e tela Inicial
- 2 Menu de cadastros
- 3 Cadastrar novo serviço a ser prestado
- 4 Financeiro

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Hoelesen/LavaCa>

Essa wiki é a documentação para uso do projeto Lava\_car, projeto desenvolvido como tese de final de curso e solução para empresas locais. para acessar ao lado direito tem os guia por páginas.

Fonte: Elaborado pelo autor, 2022


## 2- Tela Inicial do Software.

Figura 20 – Tela inicial

após efetuar o login será redirecionado para uma página com as opções da aplicação.

ID	Nome	Veículo	Hr. busca	Hr. entrega	Busca	Entrega
4	João pedro da Silva	KHH1354	15/11 09:26	16/11 12:00	<input type="checkbox"/>	<input type="checkbox"/>
5	Hoelesen França	CLM8525	17/11 10:30	17/11 11:30	<input type="checkbox"/>	<input type="checkbox"/>
6	João pedro da Silva	KHH1354	15/11 08:30	15/11 09:30	<input type="checkbox"/>	<input type="checkbox"/>
7	Pedro Santiago	MJJ-000	15/11 09:34	15/11 10:34	<input type="checkbox"/>	<input type="checkbox"/>

Os menus são para facilitar os processos.

Essa wiki é a documentação para uso do projeto Lava\_car, projeto desenvolvido como tese de final de curso e solução para empresas locais. para acessar ao lado direito tem os guia por páginas. 

Fonte: Elaborado pelo autor, 2022

### 3- Tela de cadastros.

Figura 21 – Tela de cadastros

## 2 Menu de cadastros

Hoelesen França edited this page 6 days ago · 1 revision

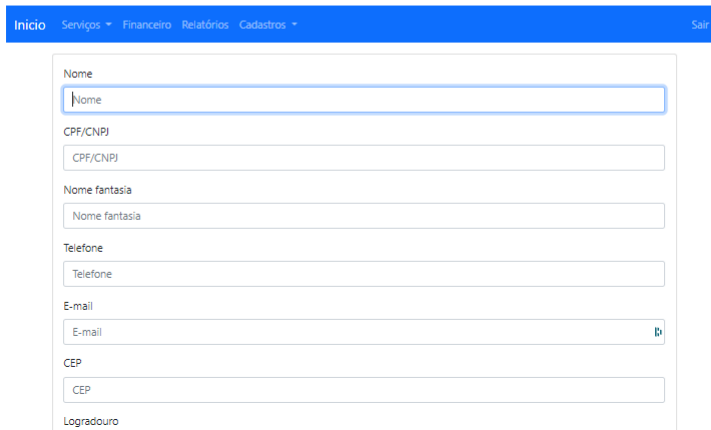
Edit [New page](#)

Na pagina inicial tem o menu cadastros no qual se é necessário preencher para criação de um novo serviço.

1- Cadastro de clientes. Nessa tela é possível cadastrar novos clientes e visualizar os clientes já cadastrados. Na opção decadastro.



CPF/CNPJ	Nome	Nome fantasia	Telefone	Ações
00000000000	Pedro Santiago	Confecções pedro	69 9 0000-0001	
85974765249	Hoelesen França	Japa	00000000	
00000000000	João pedro da Silva	Pedro Industria	00 0 0000-0000	
00000000000	Roberto Souza	Roberto Importações	69 9 00000000	



Nome

Nome

CPF/CNPJ

CPF/CNPJ

Nome fantasia

Nome fantasia

Telefone

Telefone

E-mail

E-mail

CEP

CEP

Logradouro

Pages 5

Find a page...

- ▶ 0 Bem vindo a Lava Car
- ▶ 1 Login e tela Inicial
- ▶ 2 Menu de cadastros
- ▶ 3 Cadastrar novo serviço a ser prestado
- ▶ 4 Financeiro

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Hoelesen/LavaCa>

Fonte: Elaborado pelo autor, 2022

## 4- Tela de todos serviços.

Figura 22 – Todos os serviços

























Seguindo em serviço, a opção todos os serviços é possível visualizar o serviço cadastrado nele temos quatro ícones ao lado esquerdo na coluna ações, o ícone com o símbolo de exclamação contém um relatório de alterações no pagamento e funcionários que fizeram o serviço, o ícone do lápis é a opção de editar as informações inseridas ao criar o serviço, o ícone com o símbolo do dinheiro é a opção que se acerta os valores de pagamentos já efetuados e pendentes do serviço, e o último ícone com o símbolo da lixeira é onde se é possível apagar o serviço já criado.

Início Serviços ▾ Financeiro Relatórios Cadastros ▾ Sair

### Serviços

Novo

Status: Todos Data inicial: dd/mm/aaaa Data final: dd/mm/aaaa Filtrar

N°	Cliente	Veículo	Entrada	Saída	N° nota	Status	Ações
2	Hoelsen França	CYA5797	13 de Outubro de 2022 às 11:23	13 de Outubro de 2022 às 14:23		Finalizado	   
3	Pedro Santiago	MJJ-000	13 de Outubro de 2022 às 15:25	13 de Outubro de 2022 às 16:00		Lavagem em andamento	   
4	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 09:26	16 de Novembro de 2022 às 12:00		Aguardando lavagem	   
5	Hoelsen França	CLM8525	17 de Novembro de 2022 às 10:30	17 de Novembro de 2022 às 11:30		Aguardando lavagem	   
6	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 08:30	15 de Novembro de 2022 às 09:30		Finalizado	   
7	Pedro Santiago	MJJ-000	15 de Novembro de 2022 às 09:34	15 de Novembro de 2022 às 10:34		Finalizado	   

Essa wiki é a documentação para uso do projeto Lava\_car, projeto desenvolvido como tese de final de curso e solução para empresas locais, para acessar ao lado direito tem os guia por páginas.

Fonte: Elaborado pelo autor, 2022

## 5- Tela financeiro.

Figura 23 – Tela financeiro

## 4 Financeiro

Hoelesen França edited this page 6 days ago · 1 revision

Edit

New page

E por fim temos a tela do financeiro onde é possível filtrar os serviços já pagos e os pendives, conforme alterações na tela anterior onde se era possível editar o valor do serviço impacta totalmente no filtro efetuado, nesta tela caso falte algum valor do serviço essa tela mostrará o mesmo como pendente. Nessa tela é possível filtrar por status e por data.

Início Serviços Financeiro Relatórios Cadastros Sair

### Financeiro

Status:  Data inicial:  Data final:  Filtrar

N°	Cliente	Veículo	Entrada	Saída	N° nota	Status	Valor pendente	Valor acertado
2	Hoelesen França	CYA5797	13 de Outubro de 2022 às 11:23	13 de Outubro de 2022 às 14:23		Pago	R\$ 0,00	R\$ 50,00
3	Pedro Santiago	MJJ-000	13 de Outubro de 2022 às 15:25	13 de Outubro de 2022 às 16:00		Pago	R\$ 0,00	R\$ 30,00
4	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 09:26	16 de Novembro de 2022 às 12:00		Pendente	R\$ 108,00	R\$ 0
5	Hoelesen França	CLM8525	17 de Novembro de 2022 às 10:30	17 de Novembro de 2022 às 11:30		Pendente	R\$ 45,00	R\$ 0
6	João pedro da Silva	KHH1354	15 de Novembro de 2022 às 08:30	15 de Novembro de 2022 às 09:30		Pendente	R\$ 90,00	R\$ 0
7	Pedro Santiago	MJJ-000	15 de Novembro de 2022 às 09:34	15 de Novembro de 2022 às 10:34		Pago	R\$ 0,00	R\$ 30,00

Pages 5

- ▶ 0 Bem vindo a Lava Car
- ▶ 1 Login e tela Inicial
- ▶ 2 Menu de cadastros
- ▶ 3 Cadastrar novo serviço a ser prestado
- ▶ 4 Financeiro

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/Hoelesen/LavaCa>

È isso tenha um bom uso tanto para implantação como para uso do projeto como base para novos projetos.

Essa wiki é a documentação para uso do projeto Lava\_car, projeto desenvolvido como tese de final de curso e solução para empresas locais. para acessar ao lado direito tem os guia por páginas.

Fonte: Elaborado pelo autor, 2022



# Anexos



# ANEXO A – Licença MIT

MIT License

Copyright (c) 2021 Hoelesen França

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.