



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

SYSCAR - Sistema de gestão para concessionárias

João Paulo Ferreira Beltrame

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Alexandre Magno de Sousa

**Dezembro, 2020
João Monlevade–MG**

João Paulo Ferreira Beltrame

**SYSCAR - Sistema de gestão para
concessionárias**

Orientador: Alexandre Magno de Sousa

Monografia apresentada ao curso de Engenharia da Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Dezembro de 2020

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

B453s Beltrame, Joao Paulo Ferreira .
SYSCAR [manuscrito]: sistema de gestão para concessionárias. / Joao Paulo Ferreira Beltrame. - 2020.
67 f.: il.: color., tab.. + Fluxograma. + Diagrama.

Orientador: Prof. Me. Alexandre Magno de Sousa.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Veículos. 2. Processos de fabricação - Automação . 3. Sistemas de informação gerencial. 4. Aplicações web. I. Sousa, Alexandre Magno de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.775

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

João Paulo Ferreira Beltrame

SYSCAR - Sistema de Gestão para Concessionárias

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de graduação em Engenharia de Computação.

Aprovada em 21 de dezembro de 2020.

Membros da banca

Mestre - Alexandre Magno de Sousa - Orientador - Instituto de Ciências Exatas e Aplicadas
Doutor - George Henrique Godim da Fonseca - Instituto de Ciências Exatas e Aplicadas
Doutora - Tatiana Alves Costa - Instituto de Ciências Exatas e Aplicadas

Prof. Alexandre Magno de Sousa, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 10/01/2021.



Documento assinado eletronicamente por **Alexandre Magno de Sousa, PROFESSOR DE MAGISTERIO SUPERIOR**, em 07/05/2021, às 16:40, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0168866** e o código CRC **3CF057D8**.

Este trabalho é dedicado a minha mãe Elizabeth e minha irmã Mariana, que desde o começo acreditaram que eu podia chegar onde estou agora.

Agradecimentos

Agradeço primeiramente a minha mãe e irmã, por terem apostado no meu sonho de me formar em uma federal. A minha noiva, que fez parte da minha vida desde o início do desenvolvimento do trabalho, foi uma companheira inseparável e inabalável. A todos meus familiares que me apoiaram, meus mais sinceros agradecimentos. Vocês foram essenciais!

Aos meus companheiros de república, que passaram comigo os melhores (e piores) momentos dos meus anos em Monlevade, não teria como eu lhes agradecer mais. Em especial ao Chun-li e ao Damasceno, vocês serão meus amigos irmãos pelo resto da vida.

A minha formação e busca pelo conhecimento, dedico agradecimentos aos pós-junior que fizeram parte da Visão comigo. Foi lá que desenvolvi meu gosto por desenvolvimento web, por trabalhar em grupo e perceber que poderia fazer a diferença nesse meio. Meu segundo agradecimento vai ao LEDs, por me proporcionar toda a experiência de desenvolvimento e lançamento. Vocês são demais!

Aos demais que amigos e pessoas que fizeram partes da minha vida nesse tempo, um pedacinho desse trabalho também é mérito seus. Não sei como seria se tivesse sido diferente, só tenho a agradecer a todos.

“Se é idiota e funciona, então não é idiota.”

— Autor desconhecido

Resumo

A adoção da tecnologia da informação no ambiente de trabalho aumentou nos últimos anos, não só impulsionada pelo maior acesso a sistemas de gestão e controle, mas também pelos benefícios proporcionados em sua implementação. Através do uso destes sistemas, tarefas repetitivas que antes eram feitas de forma manual hoje podem ser automatizadas, tornando o processo de trabalho mais rápido, confiável e menos custoso. Este trabalho tem como objetivo o desenvolvimento de um sistema capaz de gerenciar vendas, além da gestão de balanços monetário e de estoque de uma concessionária que atua no estado de Minas Gerais. Após o levantamento do processo de trabalho e necessidades do usuário final, foram criados os requisitos que embasam as funcionalidades presentes no sistema proposto chamado de Syscar, desenvolvido como um Sistema Web utilizando arquitetura de micro serviços. O resultado final mostra um sistema capaz de atender as necessidades diárias de operações como cadastro de clientes, veículos e funcionários, além da possibilidade de controle do estoque e balanço monetário.

Palavras-chaves: Concessionária de Veículos. Automação de Processos. Sistemas de Informações Gerenciais. Sistema Web Syscar.

Abstract

Adopting information technology on working environment increased in the last year, and the reason it's not only explained by more access to management systems and control, but also for the benefits provided by its application. Through its usage, task characterized by repetition that used to be done by manual effort can be automatized, bringing speed, confidence and a lower cost to the working process. This final paper has as goal the development of a system capable of managing sales, besides monetary balance's management and stock from a car dealership placed on Minas Gerais' state. After checking how working pipeline used to be done and user needs, base requirements were created and are presents in the proposed system called Syscar, developed as a Web System using micro-services architecture. The final result shows a system able to answer daily operational necessities such as register of customers, vehicles and employees, besides possibility of stock control and monetary balance.

Key-words: Car Dealership, Process Automation, Management Information Systems, Web System Syscar.

Lista de ilustrações

Figura 1 – Estatísticas do Google Meu Negócio.	30
Figura 2 – Diagrama de caso de Uso.	33
Figura 3 – Diagrama de atividade.	34
Figura 4 – Diagrama ER.	35
Figura 5 – Representação do Factory no sistema.	37
Figura 6 – Diagrama de sequência.	41
Figura 7 – Diagrama de Fluxo de Dados.	42
Figura 8 – Tela de Login.	42
Figura 9 – Tela principal da aplicação.	43
Figura 10 – Cadastro de veículo.	43
Figura 11 – Cadastro de cliente.	44
Figura 12 – Cadastro de funcionário.	44
Figura 13 – Cadastro de usuários.	45
Figura 14 – Criar venda.	45
Figura 15 – Visualizar estatísticas.	46
Figura 16 – Lista de veículos.	46
Figura 17 – Lista de usuários	47
Figura 18 – Lista de clientes	47
Figura 19 – Lista de funcionários.	48
Figura 20 – Lista de vendas.	48
Figura 21 – Geração de contrato.	49
Figura 22 – Alterar senha.	50
Figura 23 – Interface do <i>Postman</i>	51
Figura 24 – Organização dos elementos no GitHub.	51
Figura 25 – <i>Issues</i> no GitHub.	52
Figura 26 – Falha ao fazer o <i>login</i>	54
Figura 27 – Campos obrigatórios ao cadastrar um veículo.	55
Figura 28 – Campos obrigatórios ao cadastrar um cliente.	55
Figura 29 – Campos obrigatórios ao cadastrar um funcionário.	56
Figura 30 – Campos obrigatórios ao cadastrar um usuário.	56
Figura 31 – Campos obrigatórios ao criar uma venda.	57
Figura 32 – Menu Configurações.	59
Figura 33 – Erro ao atualizar senha.	59

Lista de tabelas

Tabela 1 – Resumo das qualidades dos trabalhos relacionados.	28
Tabela 2 – Requisitos funcionais	32
Tabela 3 – Requisitos não-funcionais	32
Tabela 4 – Descrição da entidade Pessoa.	35
Tabela 5 – Descrição da entidade Funcionários.	36
Tabela 6 – Descrição da entidade Contatos.	36
Tabela 7 – Descrição da entidade Custos.	36
Tabela 8 – Descrição da entidade Venda.	36
Tabela 9 – Descrição da entidade Usuário.	36
Tabela 10 – Descrição da entidade Veículos	36

Lista de abreviaturas e siglas

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
BI	Business Inteligence
CMS	Content Management System
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DAO	Data Access Object
DTO	Data Transfer Object
ER	Entidade-Relacionamento
ERP	Enterprise resource planning
GB	Giga-Byte
HQL	Hibernate Query language
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JPA	Java Persistence API
JSON	JavaScrpit Object Notation
JVM	Java Virtual Machine
JWT	JSON Web Token
MB	Mega-Byte
ORM	Object-relational Mapping
PDF	Portable Document Format
RAM	Random Access Memory
REST	REpresentational State Transfer

RH	Recursos Humanos
SASS	Syntactically Awesome Style Sheets
SIG	Sistemas de Informações Gerenciais
SGBD	Sistema de Gerenciamento de Banco de Dados
SGI	Sistemas de Gestão Integrado
SI	Sistemas de Informações
SOAP	Simple Object Access Protocol
SSD	Solid-State Drive
TI	Tecnologia da Informação
UML	Unified Modeling Language

Sumário

1	INTRODUÇÃO	15
1.1	Objetivo Geral e Específicos	16
1.2	Resultados e Contribuições	16
1.3	Estrutura da monografia	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Sistemas de Informação Gerenciais e Automatização de Processos	18
2.2	Principais Tecnologias Utilizadas no Mercado	20
2.2.1	SAP ERP	20
2.2.2	TOTVS ERP	21
2.2.3	SYDLE ONE	21
2.3	Considerações Finais	22
3	TRABALHOS RELACIONADOS	23
3.1	Desenvolvimento de um Sistema Web De Gerenciamento De Conteúdo Comercial Autônomo - SIMA	23
3.2	Sistema Computacional Web para Controle de Gestão da Produção de Café - SISCOFFEE	24
3.3	ANTARES: Um Sistema Web de Consulta de Rotas de Ônibus como Serviço Público	25
3.4	GarageParts: Sistema de Controle de Consumo, Deslocamento e Manutenções de Veículos Pessoais	26
3.5	Considerações finais	27
4	METOLOGIA E DESENVOLVIMENTO	29
4.1	Descrição da Empresa	29
4.2	Definição de Requisitos	30
4.3	Diagrama de Casos de Uso	32
4.4	Diagrama de Atividades	33
4.5	Modelo Entidade-Relacionamento	34
4.6	Tecnologias Utilizadas	38
4.6.1	Backend	38
4.6.2	Frontend	39
4.6.3	Recursos Computacionais	40
4.7	Diagrama de Sequência	40
4.8	Desenvolvimento	41

4.9	Testes	50
4.10	Estrutura e Hospedagem do Código no Github	51
5	RESULTADOS	53
5.1	Login: Entrada no Sistema	53
5.2	Cadastrar	54
5.3	Operações	56
5.4	Listas	57
5.5	Configurações	58
5.6	Considerações	59
6	CONCLUSÃO	61
6.1	Limitações do trabalho	61
6.2	Contribuições	62
6.3	Trabalhos Futuros	62
	REFERÊNCIAS	64

1 Introdução

Na era da informação, empresas se deparam com constantes mudanças e da necessidade de se adaptar ao novo. Segundo a [FGV \(2020\)](#), o gasto e investimento em TI cresceu no Brasil para 8% da receita das empresas e a maior parte é pertencente ao mercado de serviços com 11,4%. Esse crescimento pode ser explicado pelo aumento do uso de Sistemas de Informação Gerenciais (SIG) e Enterprise Resource Planning (ERP) com grandes expectativas ao crescimento. A utilização de sistemas de informações tem um papel importante no mundo dos negócios. Empresas de todo o mundo vêm investindo fortemente em softwares para sistemas de informação, além de consultoria e serviços de gestão e tecnologia da informação ([LAUDON; LAUDON, 2013](#)). A finalidade desses sistemas parte desde o cadastro e controle de estoque a geração de relatório mais complexos, como traçar o perfil de compra de clientes ou projeções de vendas futuras.

Por meio dos dados inseridos e persistidos no sistema, os SIGs auxiliam o acompanhamento do nível operacional por meio do uso de relatórios estatísticos, tabelas, e gráficos com intuito de mostrar ao gestor como o setor está se saindo ([ELEUTERIO, 2015](#)). Com isso, a área gerencial da empresa recebe informações qualificadas para a criação de índices e estratégias para as tomadas de decisão nos negócios.

A falta de automatização nos processos em uma corporação pode acarretar em perda de desempenho pelo trabalho manual repetitivo, além de perpetuar tarefas com alto custo de tempo e passíveis de falha humana ([MOHAPATRA, 2009](#)). O gerenciamento uma empresa sem o auxílio de um software especializado pode se tornar uma missão complexa quando existe um fluxo considerável de clientes e produtos, o que dificulta a tomada de decisões e causa lentidão na realização de processos e atividades internas de uma empresa.

A Carvel Veículos é uma concessionária atuante no ramo de vendas, trocas e financiamento de novos e semi novos situada em Ipatinga-MG, com uma metodologia de trabalho voltada ao uso de planilhas, pastas compartilhadas sem controle de acesso e preenchimento manual de contratos. A utilização de um SIG no trabalho diário da empresa têm o potencial de transformar processos repetitivos e ausência de controle nos dados cadastrais em eficiente meio operacional, além de possibilitar a visualização de relatórios de vendas, antes feito utilizando-se dados da planilha. A partir disso, o sistema criado neste trabalho visa substituir a antiga forma de trabalho, a fim de trazer mais agilidade e confiabilidade em ações comuns e rotineiras.

1.1 Objetivo Geral e Específicos

Dentro do contexto apresentado anteriormente, tem-se como proposta a criação de um sistema de controle para concessionária que seja capaz de cobrir tarefas de controle de estoque, gerenciamento de clientes, vendas de veículos e custos e geração de relatórios de estatísticas. Para alcançar este objetivo, os seguintes objetivos específicos são definidos:

- Revisão da literatura sobre a fundamentação teórica e trabalhos relacionados;
- Levantamento e análise de requisitos e criação de histórias de usuário;
- Criação de modelo de banco;
- Pesquisa de arquiteturas para sistemas;
- Definição das tecnologias de desenvolvimento para o projeto;
- Desenvolvimento de versão;
- Realização de revisões e testes;
- Lançamento de versão.

1.2 Resultados e Contribuições

Como resultado desse trabalho foi criado o Sistema Syscar, um sistema voltado ao gerenciamento de concessionárias, que proporciona os usuários finais fazerem controle de estoque, gerenciamento de clientes e vendas, além de gestão de seus cursos. Através do sistema, faz-se possível a verificação de balanço sazonal e geração de contratos a partir das vendas cadastradas no sistema, substituindo-se a utilização de planilhas de controle e preenchimentos manuais de contratos novos, contribuindo com a evolução e crescimento do modelo de negócios da empresa.

1.3 Estrutura da monografia

Este trabalho está estruturado conforme descrito a seguir. A fundamentação teórica utilizada neste trabalho é apresentada no próximo capítulo, à qual apresenta os principais conceitos sobre sistemas de informações gerenciais e automação de processos. O Capítulo 3 aborda os trabalhos relacionados, onde é apresentada descrição de cada trabalho, suas principais características e como eles contribuíram para o desenvolvimento deste trabalho. No Capítulo 4 é apresentada a metodologia e o desenvolvimento deste trabalho, desde o levantamento e análise de requisitos até escolha de tecnologias para desenvolvimento, implementação e testes. Os resultados do produto final deste projeto são apresentados no

Capítulo 5 e, por fim, o Capítulo 6 apresenta as conclusões, descreve sobre suas limitações, as principais contribuições e os trabalhos futuros.

2 Fundamentação Teórica

Este capítulo tem como objeto a apresentação dos principais conceitos que formam os fundamentos deste projeto e também apresenta tecnologias consagradas no meio de sistemas de informações gerenciais, e uma visão geral sobre a importância das mesmas. A Seção 2.1 aborda algumas definições e conceitos, além disso, também descreve alguns e casos de sucesso voltados para transformação digital em função da aplicação de sistemas de informação gerenciais. Na Seção 2.2 são apresentados alguns exemplos de importantes tecnologias utilizadas no mercado. Finalmente, as considerações finais sobre o capítulo são apresentadas na Seção 2.3.

2.1 Sistemas de Informação Gerenciais e Automatização de Processos

De acordo com [Laudon e Laudon \(2013\)](#), um Sistema de Informação (SI) pode ser definido tecnicamente como um apanhado de componentes inter-relacionados que tem como função coletar ou recuperar, processar, armazenar e distribuir informações voltadas ao apoio de decisões, coordenação e controle de uma organização. Assim, os Sistemas de Informação Gerenciais (SIG) lidam com as questões tanto comportamentais quanto técnicas, sendo mais utilizado no nível gerencial de uma companhia como ferramenta para gestão e tomada de decisão.

Além de ser utilizado como uma plataforma para os serviços automatizados, um SIG tem como característica apresentar informações em formatos de relatórios e gráficos, utilizando dados produzidos nas operações para construí-los ([ELEUTERIO, 2015](#)). As interfaces gerenciais e operacionais muitas vezes são separadas, utilizando-se de listas de permissões para diferentes visões dentro do sistema.

A utilização dos dados operacionais para a geração de relatórios e gráficos corresponde a uma importante parte de apoio a tomada de decisões, e possui como um de seus propósitos a sua capacidade de atribuir significado aos dados sob a ótica da gestão ([ELEUTERIO, 2015](#)). Os SIGs são comumente usados na área de gestão comercial, sendo utilizados para gerenciamento de vendas e criação de relatórios acerca de faturamentos, índices de vendas e controle de estoque. Outra área comum de aplicação é a gestão de RH que, nesse caso, trabalha com dados operacionais os quais são relativos a funcionários (folhas de pagamentos, cargos, salários, dentre outros). Então, os indicadores são exibidos a partir de relatórios para facilitar o entendimento e interpretação.

O desenvolvimento tecnológico possibilita ainda mais o encontro de soluções básicas

para a sociedade e indústria, auxiliando a preservação e manutenção de toda vantagem competitiva que se possa usufruir (MOHAPATRA, 2009). Assim, automatizar um processo é tornar processos de negócios complexos automatizados, buscando-se uma transformação digital e qualidade de serviço guiada pela expertise e pelo avanço de novas tecnologias.

Em meados dos anos 70, os computadores em sua maioria eram grandes *mainframes*, controladas por equipes altamente técnicas. Porém, isso mudou após a introdução dos computadores pessoais ao cotidiano na década de 80. O uso de computadores tornou-se ainda mais comum com o acesso às redes em meados dos anos 90. Por meio desse maior acesso aos computadores, a automatização do trabalho foi cada vez mais utilizada para realização de tarefas que já eram feitas por outros trabalhadores, o que também eliminou tarefas as quais eram ineficientes e de alto custo para as empresas. Também pode-se dizer que os SI automatizaram funções corriqueiras e padrões de negócios, como contabilidade e compras, enquanto criam, paralelamente, formas mais precisas e efetivas de realizar tarefas (FILHO, 1994).

Quando voltado para a indústria de Tecnologia de Informação (TI), a automatização de processos possibilita a criação de uma camada comum de visualização de cada serviço da organização, além da automatização de mudanças e conformidades por todos os núcleos que fazem um serviço (MOHAPATRA, 2009). O seu foco principal se concentra em automação de servidores e rede, além de gerência de armazenamento. Como exemplo para outros setores, pode-se citar o setor de transporte, que utiliza uma bilheteria digital; o setor de serviços de hotéis, com reservas e agendamentos automatizados; e também no setor de educação, com condução de avaliação de forma extensiva.

Dentro de várias áreas e aplicações, a automatização de processos torna-se parte e modelo de negócio em organizações. O alcance de seus objetivos, foi descrito por Laudon e Laudon (2013) como: excelência operacional; novos produtos; serviços e modelos de negócio; relacionamento mais estreito com clientes e fornecedores; melhor tomada de decisões; vantagem competitiva; e sobrevivência. A excelência operacional é uma conquista de níveis de eficiência e produtividade, em que se propõe a realização de tarefas da melhor forma possível auxiliado pelo SI, e isso contribui também para o lançamento de novos produtos, serviços, além de modelos de negócio, que é a forma principal de descrever como a empresa produz, entrega, e vende um produto.

Além dos pontos citados anteriormente, a aproximação com os clientes e fornecedores é algo fundamental para ampliação de lucros, sendo a aproximação com o cliente uma tarefa de identificação de preferências e desejos, enquanto a aproximação com fornecedores permite melhor sincronização de fornecimentos e objetivos, permitindo menores estoques, alcançando menos custos de estocagem. A melhoria de tomada de decisões podem ser feitas quando existem detalhes de estoque, bens e serviços, além da possibilidade de se fazer previsões e relatórios. Ao alcance dos objetivos descritos, a empresa retém uma

vantagem competitiva com as outras por meio da minimização de custos e pode oferecer o mesmo produto que os concorrentes por preços menores, assim, pode-se investir em melhor qualidade para seus produtos. O investimento em SIG está também associado à sobrevivência da empresa, pelo fato de estarem fortemente ligados aos negócios digitais. Sendo assim, se uma empresa busca vantagem competitiva, ela deve alinhar a TI com os planos de negócio, e o seu princípio básico é garantir que a tecnologia sirva seu negócio e não o contrário (LAUDON; LAUDON, 2013). Um caso em que pode-se notar a importância dos SIs nos negócios é o crescimento da líder de varejo no Brasil, Magazine Luíza¹.

A marca Magazine Luíza que, impulsionada pela transformação digital (SIG, *e-commerce* e *Big Data*), aumentou seu valor de mercado mais de 70.000% nos últimos cinco anos, conforme relatado pelo CTO (FATALA, 2020). O crescimento da marca foi alinhado também ao marketing em volta da “Lu²”, que é utilizada como a personagem digital humanizada da marca. A empresa passou a investir mais em seus meios digitais, contratando times internos de desenvolvimento, trazendo para mais perto o desenvolvedor do negócio da empresa e acelerando a sua expansão no mercado de *e-commerce*.

Por meio das evoluções tecnológicas em TI, os SIGs também evoluíram de grandes sistemas em servidores locais para sistemas alocados em sua grande maioria por servidores em nuvem. O crescimento de performance dos celulares permitiu o envolvimento mais próximo de plataformas digitais no meio empresarial e na tentativa de se conectar mais com fornecedores, clientes e funcionários, as organizações passaram a utilizar mais ferramentas de conexão sociais (LAUDON; LAUDON, 2013).

2.2 Principais Tecnologias Utilizadas no Mercado

A seguir são descritas as principais tecnologias SIG existentes no mercado e que possuem aplicações de suas plataformas difundidas internacionalmente em diversos setores da economia.

2.2.1 SAP ERP

O SAP ERP é um SGI desenvolvido pela Sistemas, Aplicativos e Produtos para Processamento de Dados (SAP) e sua versão mais recente é o SAP ERP S/4HANA. O módulo Enterprise Resource Planning (ERP) é voltado para operações do usuário e contempla funcionalidades voltadas para vendas e distribuição, gerenciamento de materiais, planejamento de produção, execução e logística e controle de qualidade (BOEDER; GROENE, 2014).

¹ Magazine Luíza: <<https://www.magazineluiza.com.br/>>

² Como a Lu elevou o patamar do marketing do Magazine Luiza: <<https://www.infomoney.com.br/negocios/como-a-lu-elevou-o-patamar-do-marketing-do-magazine-luiza/>>

O sistema tem como objetivo transformar o processo de negócios, englobando-os na plataforma e oferecendo processos automatizados, além de apoio à integração com outros sistemas (seja da companhia ou não). Versões atuais oferecem suporte de *Business Intelligence* (BI) e de funções analíticas.

Como grandes casos de sucesso, apresentam-se³ grandes empresas tais como a Astro Pneumatic, Geographe, além de gigantes como Coca-Cola, Enel, Mitsui Coal e Vinci Energies. A SAP também apresenta sucesso com pequenas empresas e, para isso, conta com plano de negócio no sistema para as mesmas.

2.2.2 TOTVS ERP

O TOTVS ERP é um SGI desenvolvido pela TOTVS e, segundo a FGV (2020), tem a maior fatia de adesão do mercado que atualmente conta com 33% de adesão entre as empresas. Como plano de negócio, a empresa oferece estratégias e utilidades para o sistema, independente se o cliente é de pequeno, médio ou grande porte.

O produto apresenta várias plataformas, desde modelagem de processos, a gerenciadores de arquivos, integradores de sistemas, além de opções voltadas para o foco de *Back Office* e RH, por exemplo. Ainda são oferecidas soluções verticais em 12 (doze) setores, integradas ao *Back Office* (TOTVS, 2019).

Para o ERP, são apresentados vários casos de sucesso, entre eles de nomes conhecidos tais como a Cervejaria Theresópolis, Associação Cristã de Moços (ACM) e o time de futebol Vasco da Gama. A empresa também mostra vários outros casos de sucessos de menores carteiras e empresas de portes médios.

2.2.3 SYDLE ONE

O SYDLE ONE é um SGI *all-in-one* que busca englobar variadas soluções empresariais em um mesmo ambiente. O sistema tende a utilizar a flexibilidade do mesmo para a integração a outros sistemas, também busca concretizar transformação digital em empresas.

Como serviços oferecidos, o SYDLE ONE conta com uma plataforma de Customer Relationship Management (CRM), automatização de processos, gestão de documentos e arquivos, *analytics*, além de serviços de *service desk* e de gestões, voltados para o mercado de energia, por exemplo.

Algumas grandes empresas que são clientes: SERASA, Banco Inter, e a Petrobras. Além de vários clientes de grande porte, pode-se perceber a diferenciação de seguimentos entre os casos de sucesso e várias pequenas de pequeno porte entre eles.

³ Pesquisa avaliações e estudos de caso de clientes da SAP - <https://www.sap.com/brazil/about/customer-stories/finder.html?sort=latest_desc>

2.3 Considerações Finais

A automatização de processos começou como um diferencial de negócios e, devido à entrega e produtividade que proporciona, se consolidou como um fator essencial para competitividade e redução de custos. A utilização de SIGs reduz o tempo de trabalho perdido em processos manuais demorados e planilhas, ambos não automatizados. O caso da Magazine Luiza mostra como o crescimento e o aumento nos lucros está ligado ao uso da TI alinhado aos modelos de negócio.

A alta adesão aos softwares listados na Seção 2.2 mostra que o mercado está buscando automatizar seus processos e modelos de negócio. A adição constante de novas funcionalidades por parte das desenvolvedoras dos produtos, como *Business Intelligence* ou a criação de produtos destinados a áreas ou nichos específicos, demonstra adaptabilidade e preparo para atender demandas e necessidades dos mais variados clientes.

Soluções existentes no mercado trabalham para englobar os modelos de negócio dentro de suas plataformas, utilizando integrações para ligar diversas áreas da empresa. Os casos de sucesso mostram que a transformação digital não cabe somente às empresas com grandes negócios, mas também para pequenas contas. Isso mostra que a automatização de processos de negócio está cada vez mais presente e necessária hoje em dia, além de mais difundida apesar do grande valor de investimento.

3 Trabalhos relacionados

Este capítulo trata dos trabalhos relacionados os quais apresentam uma grande importância da aplicação de automação de processos para áreas específicas do mercado. Da Seção 3.1 a 3.4 serão apresentadas as descrições de cada trabalho relacionado, tais como seus objetivos, metodologias aplicadas e qual é o problema alvo que os autores propõem a resolver. Por fim, a Seção 3.5 apresenta as considerações sobre os pontos positivos e pontos negativos de cada trabalho e também mostra como este projeto complementa esses trabalhos como uma forma de aplicação prática de automação de processos para mais uma área do mercado.

3.1 Desenvolvimento de um Sistema Web De Gerenciamento De Conteúdo Comercial Autônomo - SIMA

O SIMA tem como objetivo principal a criação de um sistema Content Management System (CMS) para microempreendedores autônomos gerenciarem seus projetos. Assim, o projeto trás uma proposta de um sistema dinâmico de forma que as funcionalidades do sistema possam se adaptar às necessidades de cada organização. A principal ideia de desenvolver um sistema de gerenciamento dinâmico é pular a parte do processo de criação de software onde você cria a modelagem de dados, elabora algoritmos para realizar operações e a comunicação com o Sistema de Gerenciamento de Banco de Dados (SGDB), sendo este voltado para bancos relacionais (ESTÁCIO et al., 2014).

O primeiro passo definido pelo autor foi definir o fluxo do programa através de um esquema básico de componentes. Esse esquema, apesar de bem abstrato, tem como objetivo definir o fluxo de interação dos componentes no sistema. A partir de um novo gerenciamento criado pelo usuário, o sistema faz a recuperação dos meta-dados e persiste o novo gerenciamento do usuário. Então, se o usuário quiser ver algum dado desse gerenciamento, o sistema consulta o banco e retorna o dado a esse usuário. Dividindo os componentes (o que o autor define como subsistemas (ESTÁCIO et al., 2014)), foi definida a funcionalidade que cada um teria e como seria feita a comunicação entre esses componentes. Os subsistemas são definidos como 3 (três) conforme a seguir: Coleção, Gestão e Publicação (ESTÁCIO et al., 2014).

No módulo de Coleção são concentradas todas as ações necessárias para a criação ou busca de conteúdo e métodos para criação de formulários HTML. Quando o conteúdo é criado, o sistema de Gestão faz o armazenamento em um repositório, implementando um repositório que abrange todos elementos de banco de dados, recursos e métodos

de configuração. Dessa fórmula, o módulo interage com todas as camadas do *CMS* (administração, dados e fluxo de trabalho). Por meio de *templates*, o subsistema de Publicação cuida da saída dos repositórios e gera as páginas HTML, as quais utilizam os conteúdos dos repositórios.

O sistema foi construído de forma dinâmica de modo que toda sua configuração foi parametrizada em arquivos XML, bem customizável ao que o usuário final necessita. Os dados de acesso ao banco também são parâmetros, sendo a geração das tabelas feito por entidades internas do programa. Adicionalmente, conta com uma divisão de permissões, na qual somente o super-administrador tem permissão total do sistema. Ressalta-se que todas as permissões são configuradas pelo usuário. O SIMA Foi projetado para ter uma boa escalabilidade e apesar de usar uma interface WEB, não é adequado para dispositivos móveis pelo fato dos formulários de inserção e exibição não serem responsivos.

3.2 Sistema Computacional Web para Controle de Gestão da Produção de Café - SISCOFFEE

Sistema Computacional Web Para Controle de Gestão da Produção de Café (SISCOFFEE) foi desenvolvido como um trabalho para plataforma Web (CUNHA et al., 2017). Esse sistema foi projetado para que o usuário final, que, nesse caso, é um produtor rural, possa fazer acesso e uso do mesmo em qualquer lugar e em qualquer tipo de plataforma (seja *mobile* ou *desktop*). Foi assim projetado com o objetivo de alcançar um público que não tem como adquirir grandes *softwares* já existentes no mercado.

Durante o desenvolvimento do SISCOFFEE, foi feito um estudo com mais de 40 produtores de café para fazer o levantamento das funcionalidades e necessidades dos usuários. O resultado final é um sistema na modalidade Web que é versátil e responsivo, necessitando-se somente de um ponto de acesso a Internet. O sistema tem como objetivo fazer a gestão de propriedades voltadas para a cafeicultura, isso traz uma ferramenta para o comando e gestão da produção de café (CUNHA et al., 2017).

Como resultado final, o produtor poderá criar seu próprio banco de dados e gerenciar o relatório de produções dos últimos anos, além de acompanhar processos como: adubação, irrigação, aplicação de produtos, e criação de relatórios a respeito dessas etapas (CUNHA et al., 2017). A divisão das funcionalidades foi feita a partir de módulos, sendo separados em menus específicos. Então, para cada ciclo que o usuário final for utilizar, há a possibilidade da retirada de um relatório, os quais podem ser do tipo geral ou do tipo específico.

Como linguagem principal do sistema, foi escolhido o Personal Home Page (PHP) 5.4.3, com objetivo de gerenciar as páginas criadas em HTML5, CSS3 e Java Script, sendo a última utilizada para suprir as limitações do PHP (CUNHA et al., 2017). No intuito

de aplicar efeitos em tempo real à página, foi utilizada a biblioteca do JQuery¹ e para o layout da página, o Bootstrap 3². Para o banco de dados, foi escolhido o MySQL 5.6.17.

3.3 ANTARES: Um Sistema Web de Consulta de Rotas de Ônibus como Serviço Público

O trabalho de Bastos e Jaques (2010) apresenta como objetivo construir um sistema Web para busca de rotas de usuários de transporte público. Esse sistema é chamado de ANTARES e aplica como algoritmo o A*, esse sistema usa como forma principal a API do Google Maps³ (BASTOS; JAQUES, 2010). O autor fez uma pesquisa dentro da heurística que deve ser usada para a busca de caminhos dentro do Google Maps. Como definição de distância, são aplicadas Distância Euclidiana e a de Manhattan. No sistema foram desenvolvidos dois principais módulos: Administrador e Usuário (BASTOS; JAQUES, 2010). Por intermédio de padrões UML, os autores fizeram o projeto do banco de dados e do diagrama de classes.

Para a criação do projeto, o primeiro passo foi a modelagem de dados. Foi utilizado o sistema de coordenadas geográficas (latitude e longitude) para a definição de pontos de ônibus relacionando-os às ruas e com as entidades das paradas de ônibus (BASTOS; JAQUES, 2010). As paradas de ônibus também foram relacionadas entre si para a criação de uma lista encadeada. Após obter as informações e armazenar esses dados no banco de dados, a modelagem do sistema foi criada mapeando o banco de dados, além de utilizar classes específicas para a criação da árvore. Para fazer o cálculo da rota, o autor criou métodos específicos e desacoplados, que interagem com a mesma estrutura.

De acordo com Bastos e Jaques (2010), a parada de ônibus como ponto de partida é definida através da proximidade com os pontos de ônibus, por sua vez, a parada de destino definida em relação à proximidade com o ponto de destino inserido do usuário. Então, os estados iniciais são inseridos na fila de prioridade, e não permite que o sistema recomende mais de um ônibus quando não for necessário. O sistema acessa recursivamente o nó inicial através do final e, a partir disso, o usuário tem todo o trajeto que irá percorrer.

Através de testes de usabilidade com usuário, o autor constatou que o sistema foi de uso intuitivo, tendo feito um estudo com usuários finais selecionados e obteve boa aprovação dos mesmos. Também foram feitos testes unitários com uma amostra de paradas e ruas, testando-se as duas heurísticas descritas anteriormente. Também é importante mencionar que o sistema é genérico de forma que possa ser executado para qualquer cidade quando configurado.

¹ JQuery - <<https://jquery.com/>>

² Bootstrap - <<https://getbootstrap.com/>>

³ Google Maps API - <<https://developers.google.com/maps/documentation?hl=pt-br>>

O sistema foi desenvolvido utilizando Java 5, em conjunto com Java Server Pages (JSP) e Struts da Apache, uma framework que aplica o padrão Model View Controller (MVC). Como *container* de *servlets*, foi utilizado o Apache Tomcat, versão 5.0. O banco de dados relacional escolhido para esse trabalho foi o MySQL.

3.4 GarageParts: Sistema de Controle de Consumo, Deslocamento e Manutenções de Veículos Pessoais

O trabalho visa o projeto e implementação de um sistema com o objetivo do controle financeiro de gastos com veículos pessoais (JúNIOR, 2019). As principais funcionalidades abordadas e desenvolvidas foram o controle do consumo, deslocamento e manutenções preventivas de veículos, além do cadastro do mesmo e de dados relacionados ao consumo e manutenção (JúNIOR, 2019). Ainda como objetivo, Júnior (2019) propõe que a aplicação deve ser *web*, além de gratuito e responsivo, de modo a suportar acessos a qualquer *software* tipo *browser*, como celulares ou *tablets*.

Para a concepção foi feita a listagem de requisitos funcionais e não funcionais do sistema, baseado em ferramentas similares de controle que não são disponibilizadas inteiramente de forma gratuita (JúNIOR, 2019). Através dos requisitos levantados, foram criados os atores e os casos de uso, onde Júnior (2019) descreve e referencia cada requisito que foi usado para a construção do mesmo. Então, foi criado o diagrama ER, que relaciona as entidades entre elas e mostra os atributos utilizados por cada uma.

O desenvolvimento seguiu o padrão de arquitetura MVC, que visa a divisão das camadas do programa em *model view* e *controller*. Para a linguagem de programação foi escolhido o PHP, com a *framework* Laravel⁴. Para o banco de dados, Júnior (2019) escolheu MySQL e para a construção do módulo administrativo do sistema, o AdminLTE⁵.

Para utilizar o *software*, o usuário deve fazer seu cadastro no sistema e em seguida fazer o login. As principais funcionalidades são o gerenciamento, histórico de manutenções e de abastecimento dos veículos (JúNIOR, 2019). Além disso, o usuário pode gerenciar seguros automotivos e o sistema permite transferir o veículo para outros usuários do sistema, funcionalidade não presente em outros sistemas estudados pelo autor (JúNIOR, 2019). Ao final dos resultados, o autor faz uma comparação entre os sistemas que foram estudados por ele e o construído, mostrando que seu sistema tem as mesmas funcionalidades que os outros pagos.

⁴ Lavavel Docs - <<https://laravel.com/docs/5.5/releases>>

⁵ AdminLTE - <<https://adminlte.io/docs/2.4/installation>>

3.5 Considerações finais

De modo geral, todos os trabalhos uma contribuição para sua respectiva área de aplicação no sentido de automação de processos, facilitando as operações de suas respectivas comunidades de interesse (Seção 3.3, Seção 3.1, Seção 3.2 e Seção 3.4). Mesmo com implementação com a visão simples, como o SISCOFFEE, ANTARES, GarageParts, eles entregaram o que prometeram de uma forma bem acessível ao usuário final, com uma interface simples e com uma curva pequena de aprendizado no uso do sistema. Um sistema mais complexo como o SIMA necessita de experiência no uso, devido a carga de conhecimento necessária para configurar do mesmo.

A Tabela 1 mostra os pontos positivos e negativos de cada trabalho, além de suas tecnologias e áreas de atuação. Como comentado anteriormente, o SIMA é um sistema de grande complexidade em sua arquitetura de software, além de robusto e bem projetado. O SIMA trás toda a customização que um *CMS* tem, deixando com o usuário a criação de tabelas e regras de negócio, adaptando facilmente ao que o usuário precisa gerenciar. Vale ressaltar que no SIMA, como ponto negativo, a separação da criação das páginas em dois subsistemas conforme descrito na Seção 3.1.

Os trabalhos ANTARES, SISCOFFEE e GarageParts têm como principal foco interface gráfica amigável ao usuário, enquanto que o ANTARES trouxe uma boa adaptabilidade e testes unitários de suas APIs. A contribuição desses três trabalhos foi exatamente entregar o que o usuário final precisa, passando por procedimentos de padrões de software, tais como o levantamento de requisitos, testes caixa preta e unitários com o resultado final pronto para o teste do usuário.

Tabela 1 – Resumo das qualidades dos trabalhos relacionados.

Trabalho	Pontos Positivos	Pontos negativos	Tecnologias utilizadas	Áreas de atuação
Gerenciamento De Conteúdo Comercial Autônomo	Customização, adaptabilidade	Respositividade, divisão do <i>view</i> em dois subsistemas	PHP, MySQL, HTML e XML	Gestão comercial
Gestão Da Produção De Café	Interface simples, responsividade	Tecnologia antiga	PHP, MySQL, JQuery e HTML5	Gestão de produção de café
GarageParts	Interface amigável e responsiva, funcionalidades avançadas	Ausência de testes	PHP, Laravel, MySQL, AdminLTE	Gerenciamento de gastos veiculares
ANTARES	Customização, adaptabilidade, interface simples, testes unitários	Limitação de pontos pela API, pesquisa de ruas, não-amigável para <i>mobile</i>	Java, Google Maps API, MySQL	Apoio ao usuário de ônibus

4 Metodologia e Desenvolvimento

Este capítulo tem como objetivo descrever como foi feito o projeto do sistema a partir de seu desenvolvimento, apresenta os requisitos do sistema, diagramas, modelagens, tecnologias utilizadas e recursos computacionais necessários. O método de desenvolvimento utilizado foi o cascata, que visa um desenvolvimento dirigido por planos, na qual se planeja todas as etapas do processo antes de começar de fato a desenvolver (SOMMERVILLE, 2011). Auxiliado com a ferramenta do ZenHub¹, a qual classifica toda as *issues* do GitHub² no quadro que, por sua vez, foi utilizado como versionador do sistema. Todos os requisitos funcionais foram separados em *issues* no GitHub e, conseqüentemente, em *branches*. Assim, quando finalizados, o *merge* era feito com a *branch* principal, nesse caso, a *master*.

Antes de falar sobre o desenvolvimento, a Seção 4.1 apresenta a Carvel, seu histórico e método de trabalho. Os requisitos são descritos na Seção 4.2 e os diagramas de casos de uso, atividade, ER e sequência são explicados nas seções 4.3, 4.4, 4.5 e 4.7 respectivamente. As tecnologias apresentadas para este trabalho são abordadas na Seção 4.6 e o desenvolvimento desse trabalho na 4.8. Os testes são explicados na Seção 4.9 e na Seção 4.10 é mostrado como foi estruturado o GitHub.

4.1 Descrição da Empresa

Nascida como uma sociedade chamada Tizé Veículos na cidade de Ipatinga-MG, a partir de 2002, a empresa foi assumida pelo atual (e único) proprietário, Carlos de Carvalho Pereira, passando a se chamar Carvel Veículos. Desde 2016, a Carvel opera em duas unidades e conta com oito colaboradores, atuando no mercado de venda, troca e financiamento de veículos 0km, seminovos e usados de boa procedência do estado de Minas Gerais. A empresa não trabalha com veículos sinistrados, recuperados de furto, chassi remarcado, com perda total ou qualquer eventualidade de leilão.

Em relação aos recursos computacionais, cada loja possui um computador *desktop* (padrão Intel Core i5, 8GB de memória RAM, dispositivo de armazenamento SSD 120GB) e um notebook, ambos com acesso à Internet (fibra ótica de 50Mb), os quais servem para executar a formalização das vendas, o processo de financiamento de veículos, controle de estoque, emissão de notas fiscais, contratos de compra e venda, divulgação e atualização das redes sociais com os anúncios dos veículos e também de promoções.

Atualmente, é utilizada uma planilha no Microsoft Excel para o controle de entrada

¹ Zenhub - <<https://www.zenhub.com/>>

² GitHub - <<https://github.com/>>

e saída de veículos e seus valores, essa planilha também é utilizada para a emissão de nota fiscal. O contrato de compra e venda é feito no Microsoft Word com um modelo pré-definido no qual são incluídos os dados do veículo e do cliente. No final do mês, é feito no Microsoft Excel o relatório de vendas com o intuito de apurar quantos veículos foram vendidos, quais foram vendidos e quem vendeu, além de outros dados inerentes a cada venda realizada.

Para mensurar o alcance que a Carvel tem, foram utilizados dados do Google Meu Negócio³, que é uma ferramenta para promover perfis de negócio na busca do Google ou no Google Maps. As estatísticas apresentadas na Figura 1 correspondem ao período de 15/11/2020 a 13/12/2020, onde a empresa teve 19,8 mil visualizações, sendo 10,7 mil por pesquisas no Google e 9,07 mil no Google Maps. Em relação ao número de pesquisas, foram realizadas 3 mil pesquisas diretas (quando o cliente pesquisa diretamente a Carvel) e 11,7 mil descobertas (quando o cliente faz pesquisas relacionadas e encontra a empresa). Já em termos de atividade é possível observar que o site⁴ (utilizado como mostruário virtual) recebeu 300 visitas encaminhadas do Google, 209 direcionamentos de chamadas para o telefone da empresa, 73,3 mil visualizações de fotos no Google e 40 solicitações de rota tanto no Google quanto no Google Maps.

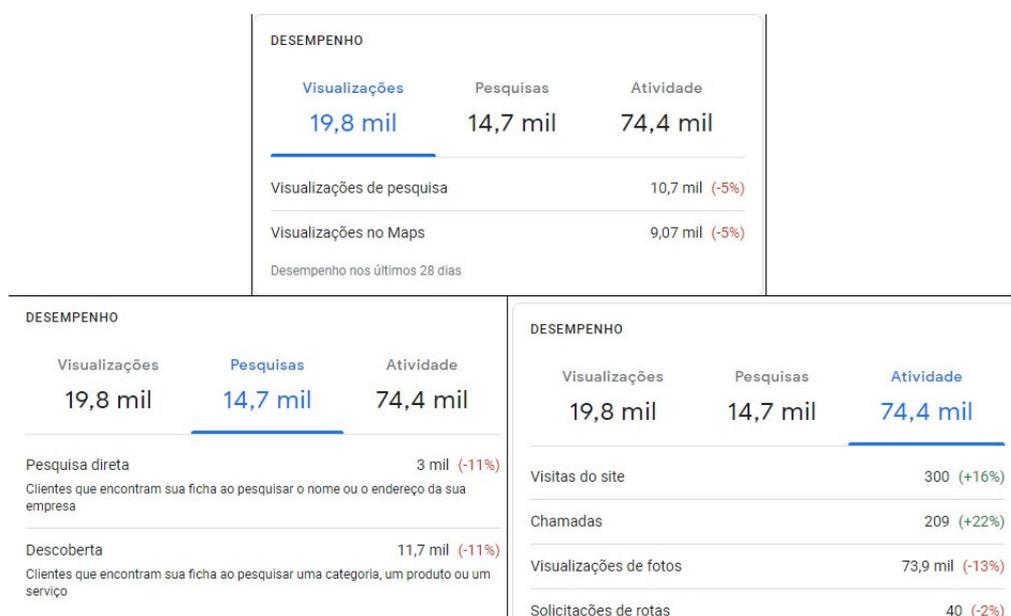


Figura 1 – Estatísticas do Google Meu Negócio.

4.2 Definição de Requisitos

Os requisitos foram definidos em uma reunião com o usuário final no início do projeto e, ao longo de outras reuniões, os requisitos foram modificados e adaptados para melhor

³ Google Meu Negócio - <https://www.google.com/intl/pt-BR_br/business/>

⁴ Carvel Veículos - <<http://www.carvelveiculosipatinga.com.br/>>

encaixar ao dia a dia de trabalho da empresa. Após entender e moldar as necessidades do usuário, os requisitos foram separados em funcionais e não funcionais. Os requisitos funcionais são descritores do que o sistema vai fazer e apresentam sua dependência ao tipo de software, seus possíveis usuários e da abordagem geral. Por sua vez, os requisitos não funcionais não estão associados diretamente aos serviços que serão oferecidos pelo sistema, além de poder se referir a restrições de implementação, capacidades de dispositivos entrada e saída (E/S) ou representação de dados na interface (SOMMERVILLE, 2011).

Os requisitos funcionais (vide Tabela 2) também podem ser considerados como as operações que os usuários vão fazer no sistema. Do requisito RF01 ao RF05, pode-se observar o termo CRUD, cuja as iniciais refletem as quatro operações básicas do banco de dados: *Create, Read, Update, Delete*. Em um banco de dados SQL, essas operações são conhecidas como INSERT, SELECT, UPDATE e DELETE, respectivamente. A instrução SELECT é a forma básica de se recuperar uma informação do banco, enquanto INSERT é usado para acrescentar uma única tupla a uma relação. Já o DELETE remove tuplas de uma relação, enquanto o UPDATE modifica valores de atributos de uma ou mais tuplas selecionadas (ELMASRI; NAVATHE, 2011).

As principais entidades existentes envolvidas no processo de negócio da empresa são: cliente, funcionário, veículo venda e usuário. Juntamente com o cadastro das entidades, o sistema também será responsável pela remoção e listagem dos mesmos. Por meio de uma venda, o usuário poderá gerar o contrato e, com a diferença do valor da venda e da compra do veículo, é feito o balanço monetário, enquanto o balanço de estoque se dá com os veículos que entraram ou saíram da loja.

A Tabela 3 mostra os requisitos que não serão operações no sistema, mas são necessidades do usuário em relação ao mesmo. Como o sistema será usado por duas lojas diferentes, há a necessidade de suportar múltiplos usuários no mesmo e, como será implementado na rede, deve-se gerar uma versão *standalone* do mesmo para ser instalado em um servidor local. Além do sistema ter a necessidade de ser construído de uma maneira que ele seja fácil de utilizar, é desejável que os valores devem vir formatados com o padrão brasileiro (tanto monetários quanto numéricos), também deve ser possível realizar a geração do contrato em arquivos de extensão Portable Document Format (PDF).

Tabela 2 – Requisitos funcionais

Código	Requisito
RF01	CRUD Cliente
RF02	CRUD Funcionário
RF03	CRUD Veículo
RF04	CRUD Usuário
RF05	CRUD Venda
RF06	Gerar contrato
RF07	Visualizar balanço monetário sazonal
RF08	Visualizar balanço de estoque sazonal
RF09	Alterar senha
RF10	Login e Logout

Tabela 3 – Requisitos não-funcionais

Código	Requisito
RN01	Acesso simultâneo de dois usuários
RN02	Ter uma versão <i>standalone</i>
RN03	Sistema fácil de usar, excluindo necessidade de treinamento
RN04	Valores devem ser formatados como padrão brasileiro
RN05	Geração do contrato em PDF

4.3 Diagrama de Casos de Uso

Os casos de uso são documentados por um diagrama de casos de uso de alto nível, que é uma característica fundamental da Unified Modeling Language (UML). Em sua forma mais simples, ele identifica os atores envolvidos em uma interação, dando nome a essa interação, suplementando-as com informação adicionais que descrevem a interação com o sistema (SOMMERVILLE, 2011). Na Figura 2 são apresentados os casos de uso desenvolvidos no sistema, sendo que existem dois atores: Administrador e Operador. De forma básica, o Operador irá fazer toda a operação do sistema no que diz respeito à realização de cadastros, geração do contrato e visualização dos relatórios. Enquanto que a figura do Administrador é responsável por gerenciar os usuários, seja adicionando ou removendo, quanto distribuindo permissões.



Figura 2 – Diagrama de caso de Uso.

4.4 Diagrama de Atividades

O diagrama de atividade tem como finalidade representar um processo e o fluxo de controle presente no sistema que o usuário ou o próprio sistema deve executar para chegar a um objetivo, explicitando as atividades e suas dependências (SOMMERVILLE, 2011). A Figura 3 demonstra o fluxo necessário para a geração de um contrato, sendo que seu caminho mais curto quando a venda já está cadastrada no sistema.

Caso a venda não esteja cadastrada no sistema, o usuário tem como pré-requisito o cadastro de um cliente, um funcionário e um veículo para operar a venda. Seguindo o fluxo de criar uma nova venda, o usuário tem como opção fazer o cadastro de cada um dessas entidades, ou utilizar as já persistidas no sistema. Após a operação, o usuário é redirecionado para a “Lista de vendas”, na qual a partir de uma venda cadastrada poderá gerar seu contrato.

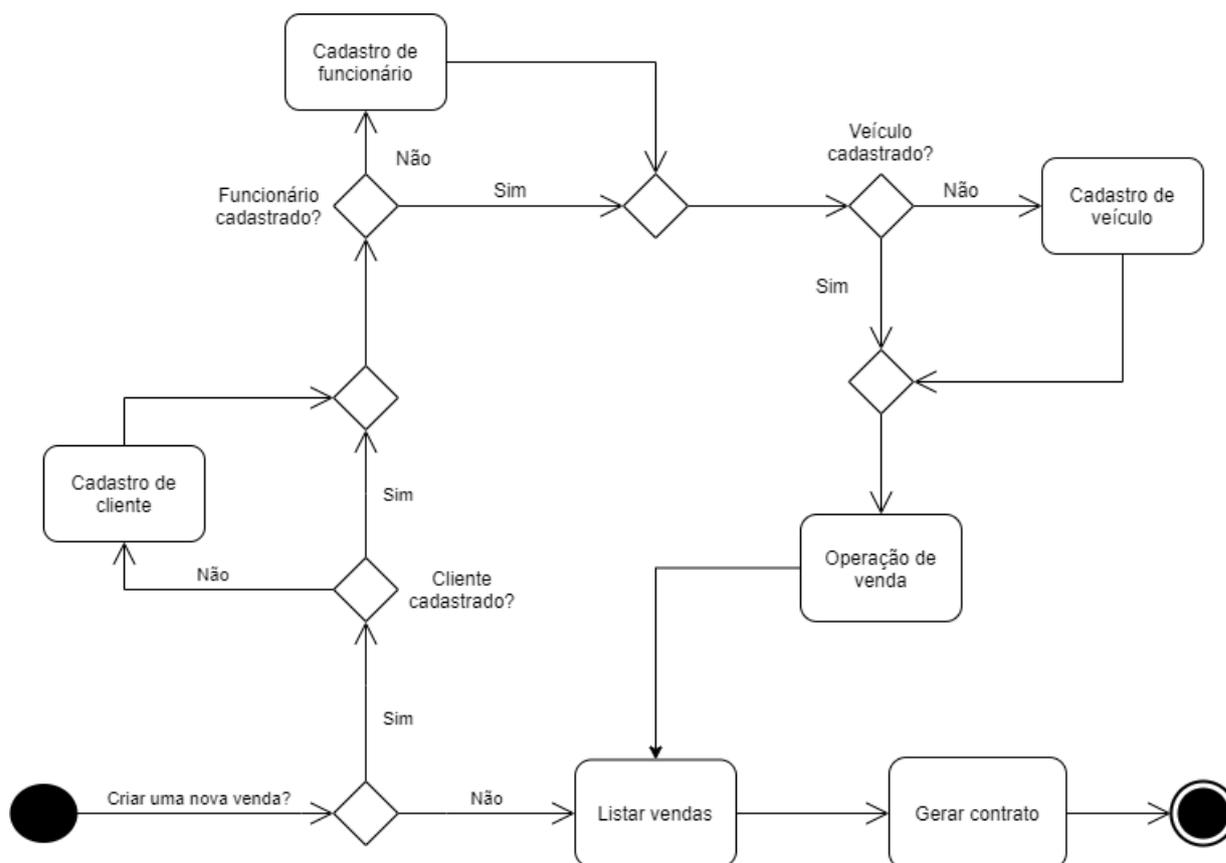


Figura 3 – Diagrama de atividade.

4.5 Modelo Entidade-Relacionamento

O modelo Entidade-Relacionamento (ER) é um modelo conceitual de alto nível, que costuma ser utilizado para o projeto conceitual de aplicação de banco de dados, além de muitas ferramentas de projeto de banco de dados utilizarem desse conceito para auxílio na construção do banco de dados (ELMASRI; NAVATHE, 2011). O modelo ER descreve os dados como entidades, relacionamentos e atributos, sendo entidades o objeto básico com uma existência independente. As entidades possuem atributos e esses tem valores cuja função é descrever essas entidades. Já os relacionamentos são uma descrição de dependências e referências entre as entidades.

A Figura 4 faz a representação do modelo ER implementado pelo sistema. A entidade Pessoa é herdada pelas outras entidades Funcionário e Cliente, e não podem coexistir em ambas instâncias. O cliente, pode ter múltiplos ou nenhum contato, assim como pode estar associado a nenhuma ou múltiplas vendas. Uma venda precisa, necessariamente, de um funcionário e um produto, além de um cliente. Finalmente, o produto pode ter múltiplos custos. A tabela de Usuário é utilizada pela parte de autenticação do programa, registrando também as permissões.

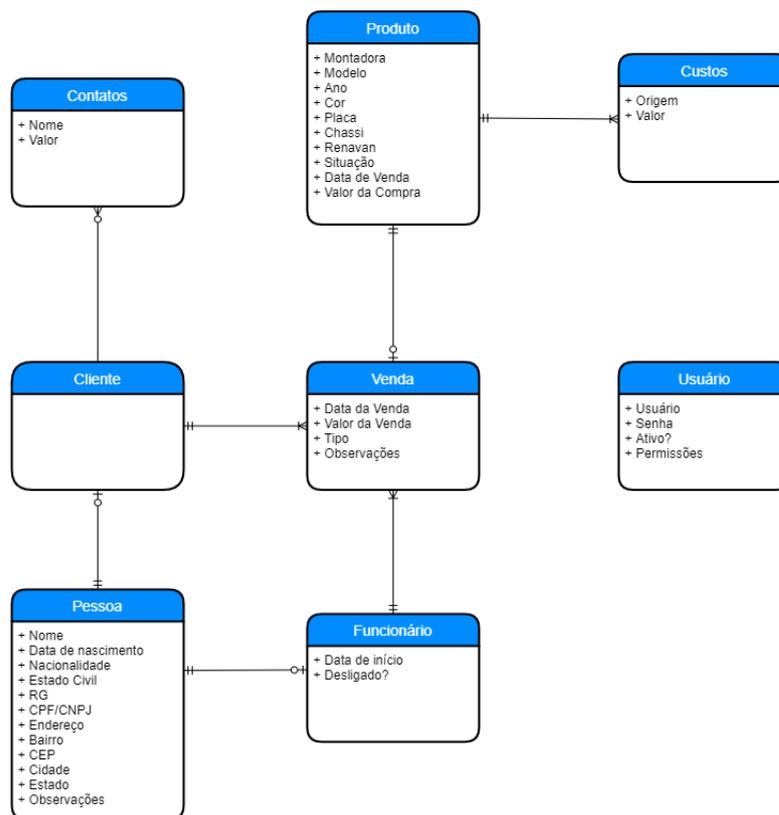


Figura 4 – Diagrama ER.

A Tabela 4 mostra todos os atributos herdados por outras duas entidades: Cliente e Funcionário, sendo a Tabela 5 explicando os campos presentes em funcionário. A Tabela 6 descreve os atributos de sua entidade e sua multiplicidade se explica pelo fato de ter mais de um meio de contato para um cliente, assim como a Tabela 7 demonstra os custos múltiplos para a compra de um veículo pela concessionária. Os campos da entidade Produto são mostrados na 10 e a Tabela 9 mostra a estrutura do usuário para o *login*.

Tabela 4 – Descrição da entidade Pessoa.

Nome do atributo	Significado	Tipo de valor
Nome	Nome da Pessoa	String
Data de nascimento	Data de nascimento da pessoa	Data
Nacionalidade	País de nascimento	String
Estado civil	Estado civil conforme a lei	String
RG	Registro da carteira de Identidade	String
CPF/CNPJ	CPF ou CNPJ	String
Endereço	Rua, logradouro e número	String
Bairro	Bairro	String
CEP	Número postal	String
Cidade	Cidade	String
Estado	Estado	String
Observações	Observações sobre a pessoa cadastrada	String

Tabela 5 – Descrição da entidade Funcionários.

Nome do atributo	Significado	Tipo de valor
Data de início	Quando o funcionário iniciou na empresa	Data
Desligado?	Se o funcionário ainda está na empresa	Booleano

Tabela 6 – Descrição da entidade Contatos.

Nome do atributo	Significado	Tipo de valor
Nome	Nome do meio de contato	String
Valor	Contato	String

Tabela 7 – Descrição da entidade Custos.

Nome do atributo	Significado	Tipo de valor
Origem	Para onde foi o gasto	String
Valor	Total gasto com essa origem	Decimal

Tabela 8 – Descrição da entidade Venda.

Nome do atributo	Significado	Tipo de valor
Data da venda	Quando o veículo foi vendido	Data
Valor da venda	Valor da venda do veículo	Decimal
Tipo	Venda ou troca	String
Observações	Observações acerca da transação	String

Tabela 9 – Descrição da entidade Usuário.

Nome do atributo	Significado	Tipo de valor
Usuário	Usuário do sistema	String
Senha	Palavra chave	String
Ativo?	Se o usuário está ativo	Booleano
Permissões	Lista de permissões do usuário	String

Tabela 10 – Descrição da entidade Veículos

Nome do atributo	Significado	Tipo de valor
Montadora	Montadora do veículo	String
Modelo	Modelo fabricado	String
Ano	Ano do modelo/ano do lançamento	String
Cor	Cor do veículo	String
Placa	Placa do veículo	String
Chassi	Chassi do veículo	String
Renavam	Renavam do veículo	String
Situação	Qual a situação dos documentos	String
Data de Compra	Data de compra do veículo	Data
Valor de Compra	Valor gasto no veículo	Decimal

A representação dessas entidades em classes foi construída com base no padrão Java Persistence API (JPA), no qual cada entidade é representada em um *bean*, a relação da entidade Pessoa para Cliente e Funcionário como uma herança e as demais relações como variáveis, onde sua multiplicidade é explicitada em cada uma delas. Todas essas entidades implementam uma interface serializável, com método de acesso ao *id*, que é a chave principal de todas as tabelas do banco. Para encapsular e abstrair a forma de leitura do banco, foi utilizado o padrão Data Access Object (DAO), que visa abstrair e encapsular os métodos e valores, além de funcionar essencialmente como adaptador entre o componente e a fonte de dados (ORACLE, 2002). A partir disso, foi utilizada uma estratégia tipo *Factory*, que é um padrão criacional e que, por definição, visa definir uma interface para criar um objeto, mais as subclasses que decidem a qual classe instanciar, assim eliminaa necessidade de anexar classes específicas de aplicações ao código (GAMMA et al., 2006).

A Figura 5 mostra como o padrão foi aplicado ao sistema, a interface *CustomRepository* é a *Factory* do projeto, e *Repository1* e *Repository2* os responsáveis pelos seus respectivos DAOs. Assim, cada entidade recebe uma interface responsável pela persistência e operações do banco de dados e o conversor para fazer a conversão dos dados persistidos para Data Transfer Object (DTO). Então, uma classe de serviço é responsável por gerenciar tanto o repositório quando o conversor, além de adicionar métodos que contêm as regras de negócio do sistema.

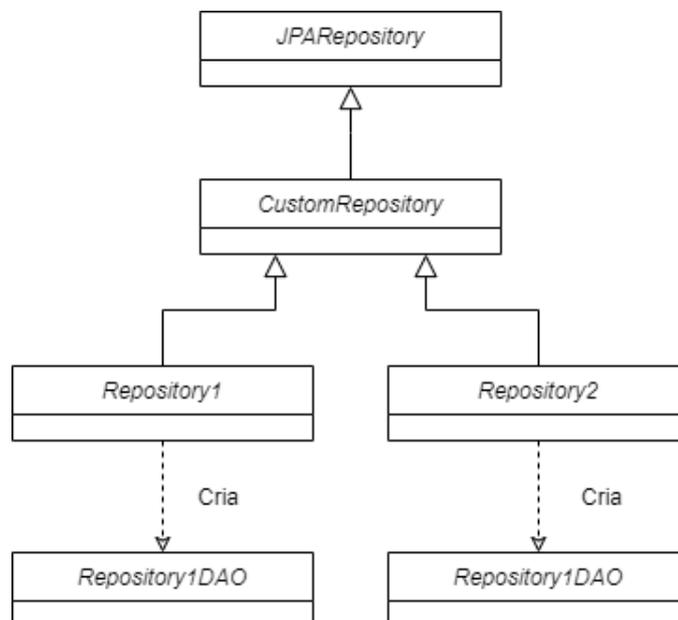


Figura 5 – Representação do Factory no sistema.

4.6 Tecnologias Utilizadas

Esta Seção traz as descrições do *backend* e do *frontend* definidos para o projeto do sistema, além disso, também traz a especificação das configurações necessárias para execução do sistema em termos de recursos computacionais.

4.6.1 Backend

A linguagem de programação escolhida para o desenvolvimento foi Java 8⁵, e sua escolha se deve à familiaridade com a mesma, além de ser uma linguagem multi-paradigma, robusta e multiplataforma. Também tem uma grande comunidade e, segundo a Redmonk Programming, é a terceira maior comunidade de desenvolvedores⁶. O ecossistema Spring⁷ foi escolhido como o conjunto de *frameworks* para se desenvolver e os recursos utilizados foram:

- O Spring Boot⁸ utiliza um pequeno conjunto de bibliotecas de terceiros e requisita somente poucas configurações, seu objetivo é a criação de uma aplicação *standalone* compacta;
- Spring Data JPA⁹ é um módulo que visa auxiliar e simplificar a implementação de repositórios baseados em JPA. Dentro do projeto foi utilizado principalmente para criação de classes de busca com padronização de *queries*;
- Como *framework* de autenticação foi utilizada o Spring Security¹⁰. Ele permite fazer a autenticação através de dados no banco, além de permitir a customização de permissões na API através de *roles*. O Auth0 foi utilizado para construir a parte de criptografar/descriptografar senhas dos usuários usando SHA256, um algoritmo de encriptação criado pela NSA que utiliza 6 funções lógicas de encriptação, cada uma operando em palavras de 32 bits (NIST, 2002).

Como Object-Relational Mapping (ORM) foi escolhido o Hibernate 4¹¹, pela facilidade de se desenvolver e ser uma framework baseada em JPA. Dessa forma, a criação do mapeamento das entidades do banco de dados fica muito mais simples, além do suporte para *queries* Hibernate Query Language (HQL). Para auxiliar na conversão de DAOs para DTOs, foi utilizado o Orika Mapper, que é uma biblioteca para mapear objetos,

⁵ Java 8 - <<https://www.java.com/pt-BR/download/help/java8.html>>

⁶ Redmonk Programming Language Rankings 2020 - <<https://redmonk.com/sogradey/2020/07/27/language-rankings-6-20/>>

⁷ Spring - <<https://spring.io/>>

⁸ Spring Boot - <<https://spring.io/projects/spring-boot>>

⁹ Spring Data JPA - <<https://spring.io/projects/spring-data-jpa>>

¹⁰ Spring Security - <<https://spring.io/projects/spring-security>>

¹¹ Hibernate - <<https://hibernate.org/>>

automatizando clonagens de objetos e sendo altamente customizável, a qual funciona como um conversor, além de aplicar regras. O banco de dados escolhido para o trabalho foi o MySQL 6.3¹². Como apoio ao desenvolvimento, foi utilizado o Spring Tool Suite 4¹³, uma distribuição modificada do Eclipse¹⁴.

4.6.2 Frontend

Como *framework* e *template engine* foi escolhido o Angular¹⁵. Para cada página mostrada no trabalho foi criado um componente, que é um subconjunto de diretivas associadas a um *template* (ANGULAR, 2020). Cada componente é criado com seu *template* HTML e um arquivo CSS, que contém todos os estilos utilizados exclusivamente por ele. Como configuração, foi criado um arquivo de estilos por meio do uso da Syntactically Awesome Style Sheets (SASS)¹⁶, que é uma linguagem para auxiliar a criação de estilos.

Para o gerenciamento de componentes no angular, existem os Modules, que são entidades para organizar injeção de dependências, além de mostrar como este deve ser compilado. Foi criado um módulo com nome *SharedModule* para organizar a injeção de componentes. Componentes criados para este trabalho voltados para as mesmas entidades, como por exemplo, lista de funcionários e cadastros de funcionários, possuem uma classe que estendia a classe abstrata *DefaultService* a qual abrange 4 chamadas básicas: *findOne*, *create*, *findAll*, e *delete*. Importante mencionar que o método *create* faz a diferenciação entre *create* e *update*, e *findOne* performa a função *get*.

O Angular nativamente cria aplicações que são responsivas, podendo ser utilizadas tanto em um computador quanto em um celular. Como biblioteca de estilização foi utilizado o Angular Material¹⁷ que abrange interfaces, botões, fontes, além de estilos de textos. Para a criação de *layouts* mais complexos foi também utilizado o Flex Layout¹⁸, que permite a criação de divisões em tela mais facilmente, além de *layouts* flexíveis e responsivos.

Para o armazenamento de sessão, foi usado o ngx-webstorage¹⁹, que permite guardar todos os dados do usuário (como *token* de autenticação e permissões) sem que o mesmo precise autenticar novamente todas as vezes que abrir uma nova página. Como o validador de permissões no *frontend* foi utilizado o CASL²⁰, que faz a leitura das permissões do usuário e filtra páginas, funções e ações que aquele usuário está habilitado pra fazer. Como IDE, foi utilizado o Visual Code²¹, além de extensões voltadas para o desenvolvimento

¹² MySQL - <<https://www.mysql.com/>>

¹³ Spring tool Suite - <<https://spring.io/tools>>

¹⁴ Eclipse - <<https://www.eclipse.org/downloads/>>

¹⁵ Angular - <<https://angular.io/>>

¹⁶ SASS - <<https://sass-lang.com/documentation/syntax>>

¹⁷ Angular Material - <<https://material.angular.io/>>

¹⁸ Flex Layout - <<https://github.com/angular/flex-layout>>

¹⁹ ngx-webstorage - <<https://www.npmjs.com/package/ngx-webstorage>>

²⁰ CASL - <<https://www.npmjs.com/package/@casl/angular>>

²¹ Visual Code - <<https://code.visualstudio.com/>>

com o Type Script, HTML e CSS.

4.6.3 Recursos Computacionais

Para rodar a Java Virtual Machine (JVM), a Oracle recomenda 2GB para plataformas *Windows*, além de 500MB livres em disco²². Os requisitos recomendados para rodar o Node JS em conjunto com o Angular são: 4GB de RAM, 10GB livres em disco e Windows 10 64-bits. Então, para esse trabalho, o recomendado seria um mínimo de 4GB de RAM, 10GB de espaço livre e Windows 10 64-bits, além de conexão com a rede interna da loja.

4.7 Diagrama de Sequência

Segundo [Sommerville \(2011\)](#), os diagramas de sequência são utilizados como ferramenta principal de modelagem das interações entre os atores e objetos, além das interações entre os próprios objetos. A Figura 6 mostra como é feita a criação de uma venda por um operador utilizando-se o sistema, interagindo com três principais componentes: *frontend*, *backend* e banco de dados.

A partir da seleção da operação de venda, o *frontend* envia requisição para o *backend* de 3 listas: funcionários, clientes e veículos, sendo os veículos aqueles que ainda não foram vendidos. A partir disso, o *backend* faz requisição no banco de todos esses dados, e os retorna para o *frontend*, então sendo exibidos para o Operador. Após o preenchimento pelo Operador dos dados da venda, os mesmos são enviados para o *backend* que ao receber os converte, adicionando-os ao banco de dados. Então, o *backend* obtém a lista de todas as vendas e as retorna para o *backend*, que então é exibida ao usuário.

²² JEE8 Sdk Release Notes - <https://www.oracle.com/java/technologies/javaee/javaee8-sdk-release-notes.html#System_Requirements>

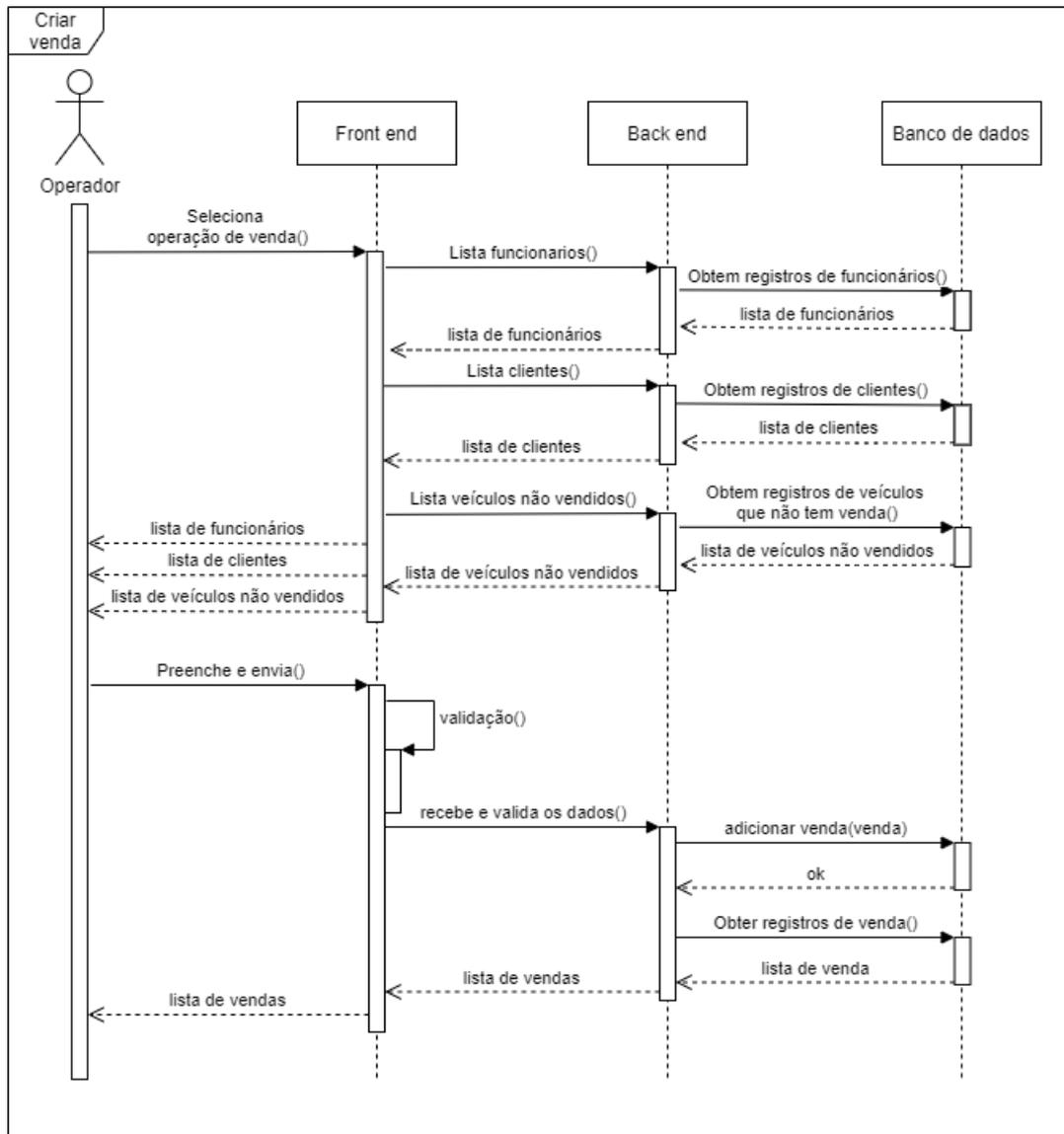


Figura 6 – Diagrama de sequência.

4.8 Desenvolvimento

Após a avaliação dos requisitos e escolha das tecnologias a serem utilizadas, o sistema começou a ser produzido. Para o *backend*, foi escolhida uma arquitetura de microsserviços baseadas em componentes e como orientação foi usado os princípios SOLID²³, recebendo uma estrutura de componentes que suporte a inclusão de novos componentes sem que se alterem os componentes já existentes dentro do serviço (MARTIN, 2019).

A Figura 7 exibe um diagrama de como o sistema interage. A partir de uma chamada do frontend (Angular), a estrutura chamada de API faz a recepção da requisição REST. As

²³ SOLID é um acrônimo para cinco postulados de design, os quais são destinados a facilitar a compreensão e o desenvolvimento e a manutenção de software: Single-responsibility Principle; Open-closed Principle; Liskov Substitution Principle; Interface Segregation Principle; e Dependency Inversion Principle

interfaces responsáveis por cada conjunto de chamadas são chamadas de Resources, e pode ser encontrado uma para cada entidade do programa. O padrão utilizado para a construção dessas interfaces foi o Factory Method, explicado na Seção 4.5. Esse conjunto de interface referencia o serviço da entidade que, ao final das operações no *service*, é convertido e persistido/consultado no banco de dados pelo DAO. Com a operação concluída, o fluxo volta até a API devolver a resposta para o *frontend*.

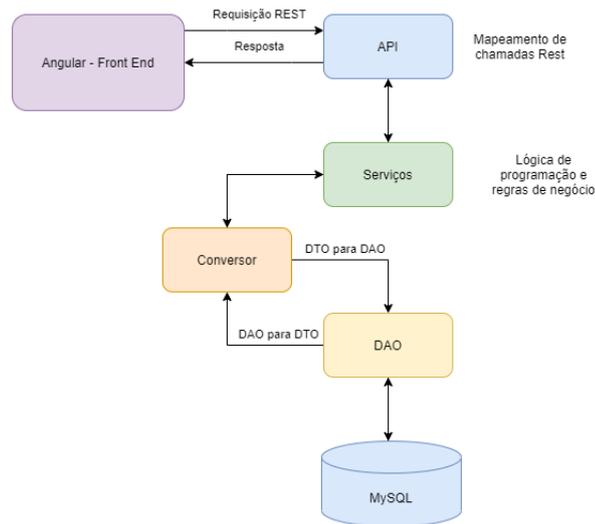


Figura 7 – Diagrama de Fluxo de Dados.

Para acessar o sistema, se o usuário ainda não estiver logado, será exibido a tela de *login* (Figura 8). Caso o usuário já tenha feito o login, a sessão é salva e o usuário seria redirecionado para a tela principal da aplicação (Figura 9). A aplicação foi dividida em três módulos disponíveis no menu: cadastros, listas, operações e configurações dos usuários, representado pelo símbolo de engrenagem.

A imagem mostra a interface de login de um sistema chamado **Syscar**. No topo, há uma barra de cabeçalho azul com o nome "Syscar" e um ícone de engrenagem. O formulário de login, intitulado "Acessar", contém dois campos de entrada: "Usuário *" e "Senha *". Abaixo dos campos, há um botão azul com o texto "Entrar".

Figura 8 – Tela de Login.

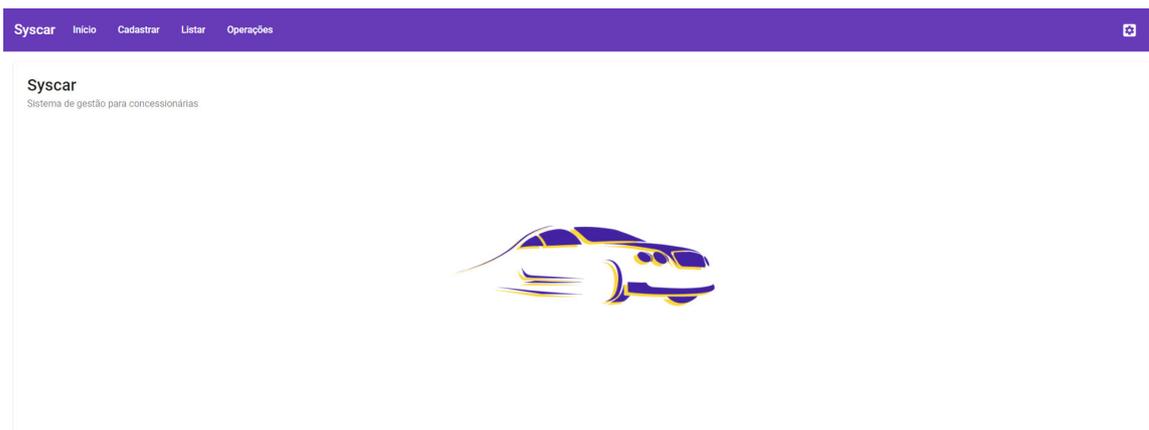


Figura 9 – Tela principal da aplicação.

No menu **Cadastrar**, se tem disponível as telas de cadastro de: funcionários, clientes e veículos. As telas foram todas criadas usando formulários reativos com a validação feita antes do envio do formulário. Foi feito o reuso do mesmo *layout* para as telas de edição. As telas de veículo, cliente, funcionário e usuário são descritas a seguir:

- Veículo: a Figura 10 mostra como é feito o cadastro de veículo. O que foi chamado de Produto na Seção 4, representa os veículos cadastrados pelo operador. Suas despesas são múltiplas e sua correspondente é a tabela de Custos;

A imagem mostra o formulário de 'Cadastro de Veículo'. No topo, há uma barra de navegação azul com o nome 'Syscar' e os menus 'Início', 'Cadastrar', 'Listar' e 'Operações'. O formulário contém os seguintes campos: 'Modelo *', 'Ano *', 'Cor *', 'Chassi *', 'RENAVAM *', 'Placa *', 'Data de compra *' (formato dd/mm/aaaa), 'Valor de compra *', 'Despesas' (com ícone de mais), 'Descrição de gasto *', 'Valor *' (com ícone de X), 'Observações' (com ícone de X) e botões 'Limpar' e 'Salvar' no canto inferior direito.

Figura 10 – Cadastro de veículo.

- Cliente: o cliente (Figura 11) tem todos os campos da tabela Pessoa e os contatos, podendo ser múltiplos e independentes de tipos;

The screenshot shows the 'Cadastro de Cliente' form in the Syscar system. The form is titled 'Cadastro de Cliente' and is located in the 'Operações' menu. It contains the following fields:

- Nome *
- CPF/CNPJ *
- RG *
- Telefone *
- Estado civil *
- Nacionalidade *
- Endereço *
- Cidade *
- Estado *
- Contatos
- Tipo de contato *
- Contato *
- Observações

At the bottom right, there are two buttons: 'Limpar' (yellow) and 'Salvar' (grey).

Figura 11 – Cadastro de cliente.

- **Funcionário:** o cadastro de funcionário (Figura 12) tem o mesmo correspondente no diagrama ER, representado na Seção 4. Contém todos os campos de pessoa e, adicionalmente, a data de início;

The screenshot shows the 'Cadastro de Funcionário' form in the Syscar system. The form is titled 'Cadastro de Funcionário' and is located in the 'Operações' menu. It contains the following fields:

- Nome *
- CPF/CNPJ *
- RG *
- Telefone *
- Data de admissão * (dd/mm/aaaa)
- Estado civil *
- Nacionalidade *
- Endereço *
- Cidade *
- Estado *
- Observações

At the bottom right, there are two buttons: 'Limpar' (yellow) and 'Salvar' (grey).

Figura 12 – Cadastro de funcionário.

- **Usuário:** administradores tem a opção de fazer o cadastro de usuários (Figura 13), correspondendo a tabela Usuário no modelo ER. As permissões também são inseridas na hora do cadastro, além de que, na edição, pode-se desativar esse usuário.

Syscar Inicio Cadastrar Listar Operações

Cadastro de usuário

Usuário * Senha *

Permissões * Ativo?

Limpar Salvar

Figura 13 – Cadastro de usuários.

A parte de **Operações** cuida da criação de vendas e estatísticas. A criação das vendas (Figura 14) é uma das funções essenciais do sistema, ela necessita que se tenha o cadastro de um funcionário, veículo e cliente já realizados previamente. Essas relações podem ser vistas no modelo ER, em conjunto com os outros campos na tabela de Venda.

Syscar Inicio Cadastrar Listar Operações

Criar Venda

Comprador * Vendedor * Veículo *

Data de venda * Tipo de venda * Valor * Valor de lucro

dd/mm/aaaa Venda 0 0

Observações *

Limpar Salvar

Figura 14 – Criar venda.

A função **Visualizar estatísticas** mostra tabelas e agregadores a partir de duas entradas: data de início e data de fim. A primeira tabela da Figura 15 apresenta os veículos que foram comprados (e ainda não vendidos), fazendo-se o total de gastos com os mesmos, que são os gastos com o valor de compra do veículo mais as despesas com o veículo. Para a tabela de Vendas realizadas, é feito a agregação dos valores gastos, da venda e do lucro, que seria o valor da venda decrescido dos gastos.

Modelo	Ano	Placa	Data de compra	Gastos totais
Corolla 2.6	2018/2019	XXX1234	07/12/2020	R\$ 58.630,00
Total				R\$ 58.630,00

Modelo	Placa	Cliente	CPF/CNPJ	Data de venda	Valor gasto	Valor da venda	Lucro
Volvo Amark 2.6 V6	XXX1819	João Paulo Ferreira Beltrame	987.654.352-78	08/12/2020	R\$ 98.837,00	R\$ 97.000,00	-R\$ 1.837,00
Total					R\$ 98.837,00	R\$ 97.000,00	-R\$ 1.837,00

Figura 15 – Visualizar estatísticas.

O menu **Listar** contém as listas de cadastros, além de listar as vendas cadastradas no sistema. Todas as listas foram feitas de forma que o registro é expandido quando clicado no mesmo, exibindo de antemão os dados mais importantes. As listas existentes são descritas a seguir:

- Lista de veículos: exibe o modelo e a placa do mesmo (Figura 16), e quando expandido mostra todos os dados do mesmo, tendo-se a opção de editar e excluir o registro;

Modelo	Placa
Volvo Amark 2.6 V6	XXX1819
Corolla 2.6	XXX1234

Figura 16 – Lista de veículos.

- Lista de usuários: como mostrado pela Figura 17, a lista de usuários mostra como prévia o *login*, e ao expandir mostra as permissões e se o usuário está ativo;

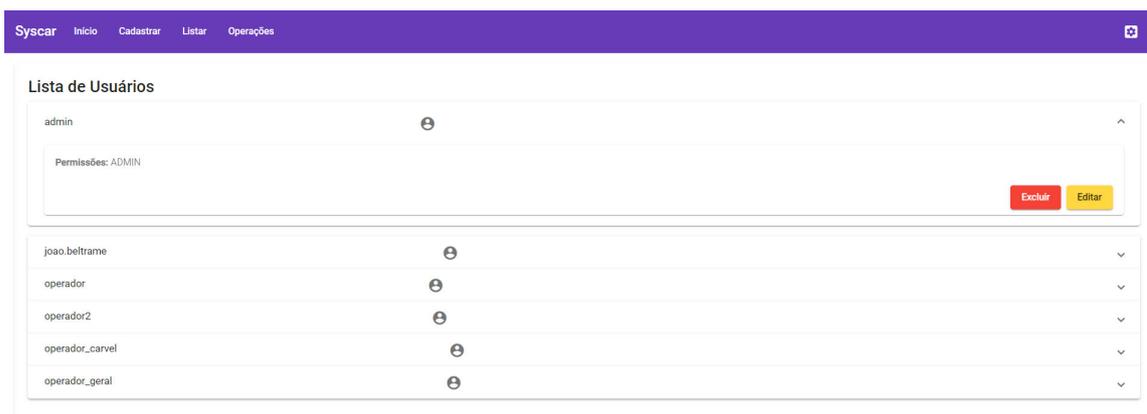


Figura 17 – Lista de usuários

- Lista de clientes e funcionários: como dados apresentados previamente, mostra o nome e o CPF do cliente (Figura 18). Também apresenta opções de editar e excluir o registro e, quando expandido, exhibe todos os dados referentes a ele. A lista é bem semelhante a lista de funcionários (Figura 19), diferenciado-se somente pelos campos padrões de cada classe;

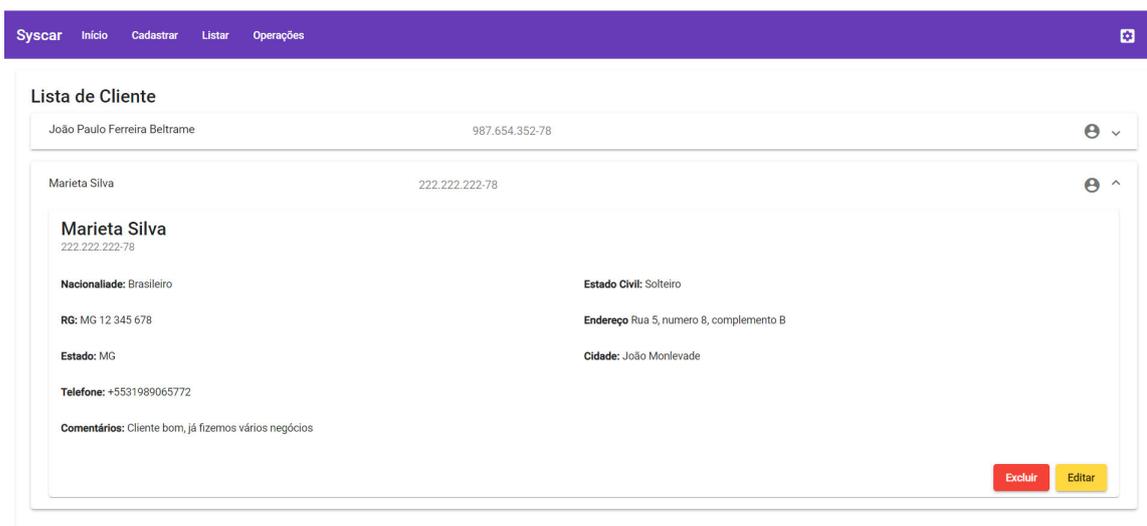


Figura 18 – Lista de clientes

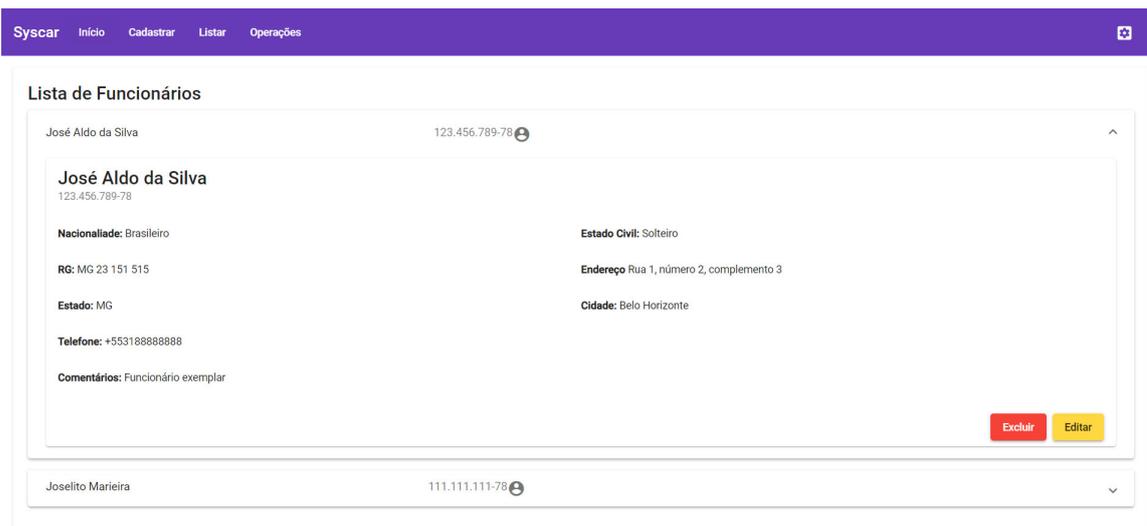


Figura 19 – Lista de funcionários.

- Lista de vendas: a lista de venda (Figura 20) contém todas informações da venda, além de algumas informações sobre o vendedor, comprador (representado pelo cabeçalho do item) e do carro. A partir do item da venda, é possível o usuário gerar o contrato com base naquela venda, clicando no item de “prancheta”. A Figura 21 mostra o contrato com os campos completados de acordo com os dados da venda e com a opção do usuário preencher dados faltantes. Tendo-se um contrato do tipo troca, o usuário tem a opção de adicionar um ou mais veículos cedidos na troca do veículo vendido. Após isso, o usuário pode transformar esse contrato em PDF.

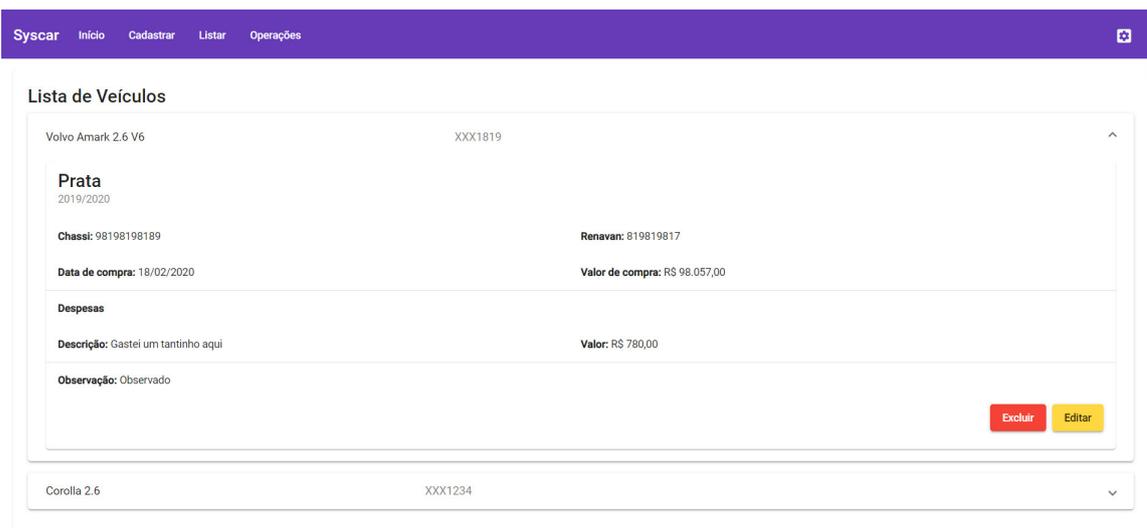


Figura 20 – Lista de vendas.




Cláusula primeira: O **CONTRATO PARTICULAR DE COMPRA E VENDA** de um veículo, que fazem de um lado como promitente vendedor **CARVEL VEÍCULOS** portador do CNPJ 04.886.445.0001/71, situada na Av. Minas Gerais, 925, bairro Jardim Panorama, cidade Ipatinga – MG. Telefone (31) 3826-8999 e do outro lado como promitente comprador (a):

Nome completo	João Paulo Ferreira Beltrame	CPF	987.654.352-78
Doc. Identidade	MG 12 345 678	Telefone	+5531989065772
Endereço	Rua 5, numero 8, complemento B	UF	MG

VEÍCULO VENDIDO/COMPRADO

Marca	Volvo Amark 2.6 V6		Ano / Modelo	2019/2020	
Chassi	98198198189	RENAVAM	819819817	Placa	XXX1819
Cor	Prata	Multa		Impedimentos	
IPVA		Licenciamento		DPVAT	

VEÍCULO CEDIDO NA TROCA

Marca		Ano / Modelo	
Chassi	RENAVAM	Placa	
Cor	Multa	Impedimentos	
IPVA	Licenciamento	DPVAT	

Entrada	Valor Financiado	Total	R\$ 97.000,00
À vista	Banco	Parcela (\$)	

OBS: ghdsafhdsfh

Cláusula terceira: O recibo original do veículo somente será entregue ao comprador após a liquidação total do(s) documento(s) acima relacionado(s).

Cláusula quarta: O comprador ficará como fiel depositário do veículo, não podendo vendê-lo ou trocá-lo enquanto o mesmo não for totalmente quitado.

Cláusula quinta: O não pagamento de quaisquer das prestações relacionadas acima nos seus vencimentos, implicará na rescisão automática deste contrato, ficando o comprador sem qualquer direito sobre o veículo e sem qualquer direito de ressarcimento da quantia já paga.

Cláusula sexta: O comprador a partir desta data fica responsável por quaisquer danos materiais ou pessoais que venham a ocorrer com o veículo. O veículo terá garantia interna de motor e caixa de 90 dias, não cobrindo a troca de peças externas, em caso do veículo nesse período apresentar defeito citado na garantia acima o comprador terá que comunicar ao vendedor imediatamente, pois somente o mesmo irá encaminhar o veículo para seu mecânico para que seja verificado tal problema, se o mecânico do vendedor detectar que tal problema foi causado por mau uso do comprador, tais como aquecimento no motor até causar danos no mesmo, a garantia perderá a validade e o comprador terá de arcar com todas as despesas causadas pelo mesmo.

Cláusula sétima: O comprador declara ter levado em mecânico de sua confiança, ter experimentado e vistoriado o veículo, portanto, o mesmo não poderá fazer qualquer reclamação a respeito do estado do veículo.

Cláusula oitava: Os veículos incluídos nas cláusulas primeira, segunda ou em ambas, se houver multas, impostos atrasados, qualquer débito que venha surgir com o veículo, que seja comprovado anterior a data e hora a contar da assinatura desse contrato, será apresentado ao vendedor citado na primeira ou segunda cláusula para que o mesmo efetue o pagamento.

Cláusula nona: Quanto ao reconhecimento da firma no recibo de transferência do veículo, o vendedor entregará no ato do negócio o mesmo com a firma reconhecida. As partes contratantes elegem o fórum de Ipatinga para todas as ações e medidas que se fizerem convenientes ou necessárias em decorrência deste contrato. E por estarem justas e contratadas, assinem o presente instrumento de contrato em duas vias de igual teor na presença de duas testemunhas igualmente abaixo assinadas.

José Aldo da Silva

João Paulo Ferreira Beltrame

Figura 21 – Geração de contrato.

Em **Configurações** (representado pela ícone de engrenagem), o usuário tem acesso a troca de senha (Figura 22), além de fazer o *logout*. Para fazer a troca de senha, o sistema confere se a *hash* da senha antiga bate, além de verificar se a senha nova e a confirmação são iguais. Após a senha persistir, o *backend* retorna o token novo e o *frontend* armazena-o na sessão.



Figura 22 – Alterar senha.

4.9 Testes

Para testar as telas no *frontend*, foram utilizados testes caixa preta, com base nos requisitos descritos na Seção 4.2. Por exemplo, o requisito RF01 - CRUD Cliente, foram testadas as interações de cadastro, listar, editar e deletar, além de formatação de campos e calores. Assim, todas as telas foram testadas previamente, bem como suas respectivas funcionalidades, de maneira a englobar todos os requisitos funcionais.

Como descrito na Seção 4.6.1, a comunicação entre o *frontend* e o *backend* foi feita por chamadas REST. Para testar essas chamadas, foi utilizado o Postman²⁴, que consegue simular chamadas para APIs, além de testar autenticações e possuir suporte para *script*, deixando os testes mais dinâmicos. Dessa forma, foi possível não só testar corpos e parâmetros das chamadas, mas também se suas saídas correspondiam ao resultado esperado.

A Figura 23 mostra um exemplo de chamada GET feita utilizando *Postman*, e mostra como se obter todos os Clientes cadastrados. O *token* JSON Web Token (JWT) é inserido manualmente por meio de uma autenticação feita previamente e a resposta é apresentado ao lado em formato JavaScript Object Notation (JSON).

²⁴ Postman - <<https://www.postman.com/>>

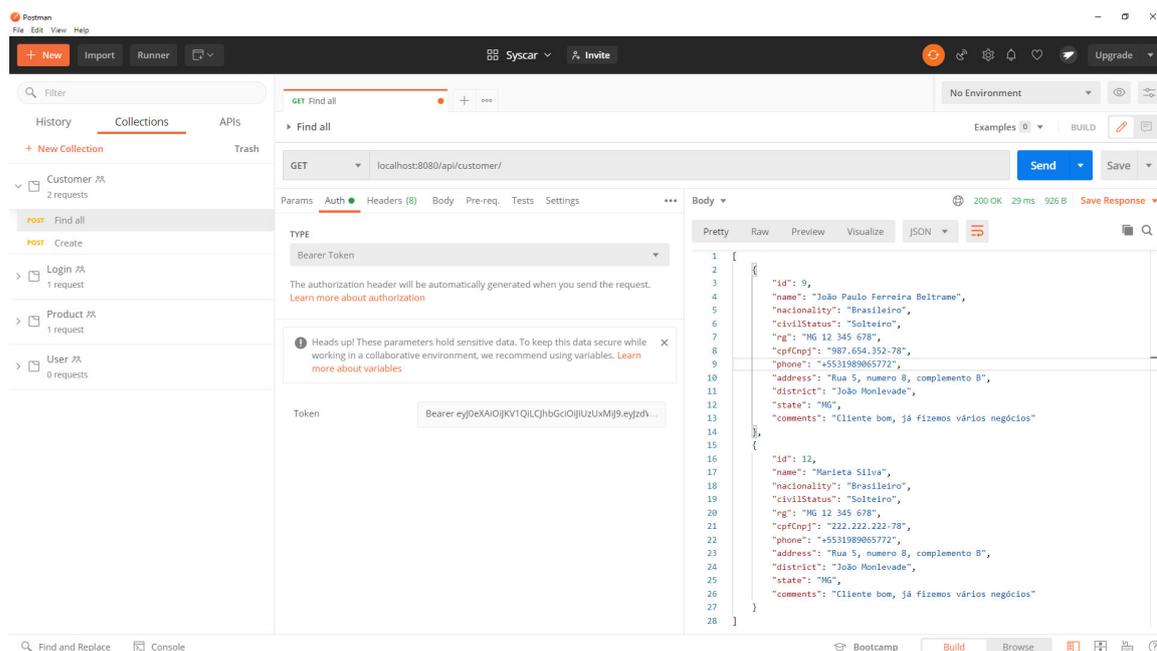


Figura 23 – Interface do Postman.

4.10 Estrutura e Hospedagem do Código no Github

O GitHub foi utilizado como versionador do trabalho, hospedando o código do *backend* e *frontend*, além do esquema do banco de dados. A Figura 24 mostra a divisão em pastas de cada parte do projeto. O `.gitignore` foi preenchido com auxílio da ferramenta `gitignore.io`²⁵, que gera, baseado nas tecnologias escolhidas, parâmetros para ignorar arquivos e pastas do projeto que não devem ser versionadas.

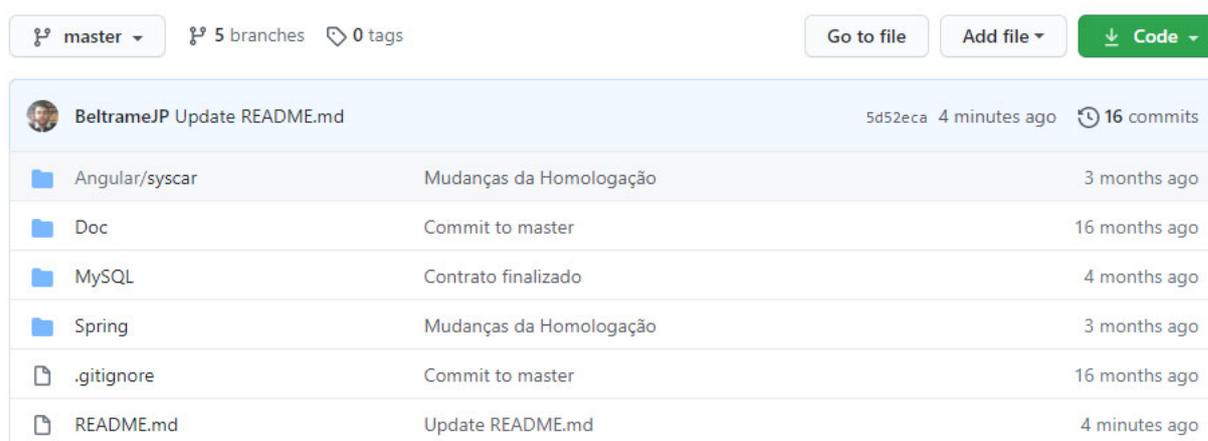
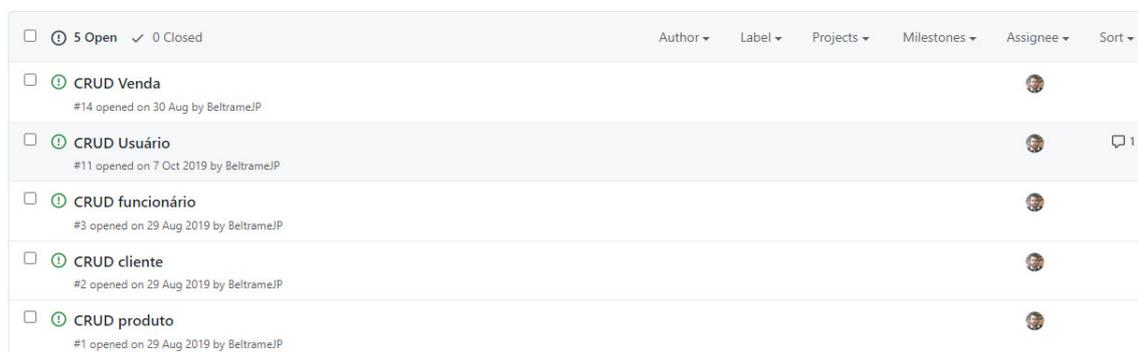


Figura 24 – Organização dos elementos no GitHub.

As *issues* do projeto (Figura 25) foram criadas com base nos requisitos apresentados na Seção 4.2, sendo que para cada uma delas uma *branch* com o padrão *is-*

²⁵ `gitignore.io` - <<https://www.toptal.com/developers/gitignore>>

`sue_#numerodaissue` foi criada. Após as *issues* serem concluídas, elas eram fundidas com a *branch master*, a principal do projeto.



The screenshot displays a GitHub Issues page with a list of five open issues. At the top, there are filters for '5 Open' and '0 Closed', along with dropdown menus for 'Author', 'Label', 'Projects', 'Milestones', 'Assignee', and 'Sort'. Each issue entry includes a checkbox, a title, a description, the number of comments, and the assignee's profile picture.

Issue Title	Description	Comments	Assignee
CRUD Venda	#14 opened on 30 Aug by BeltrameJP	0	BeltrameJP
CRUD Usuário	#11 opened on 7 Oct 2019 by BeltrameJP	1	BeltrameJP
CRUD funcionário	#3 opened on 29 Aug 2019 by BeltrameJP	0	BeltrameJP
CRUD cliente	#2 opened on 29 Aug 2019 by BeltrameJP	0	BeltrameJP
CRUD produto	#1 opened on 29 Aug 2019 by BeltrameJP	0	BeltrameJP

Figura 25 – *Issues* no GitHub.

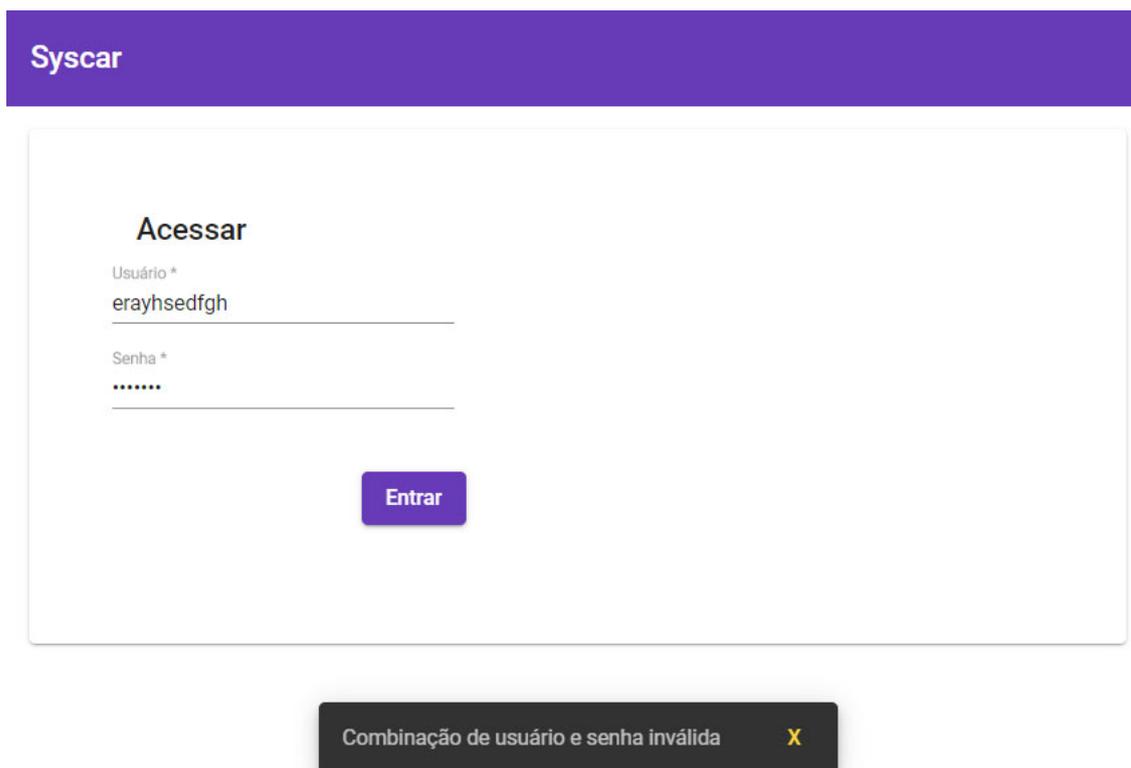
5 Resultados

Este capítulo apresenta o sistema como resultado e produto final do desenvolvimento deste trabalho, mostrando todas as ações que podem ser realizadas no sistema, além dos possíveis *feedbacks* apresentados pelo ele. Nesse contexto, serão descritas passo a passo as funcionalidades com objetivo final de demonstrar como é feita a venda de veículos e leitura das estatísticas. As seções foram divididas com base nos menus do *frontend* e, ao final, serão feitas as considerações em torno dos resultados alcançados.

A Seção 5.1 apresenta como é feito o *login* no sistema e também o *logout*. O módulo Cadastrar é apresentado na Seção 5.2, enquanto as operações são descritas na Seção 5.3. Já o módulo de listas é descrito na Seção 5.4 e as configurações de conta são apresentadas na Seção 5.5. As considerações finais acerca dos módulos serão feitas na Seção 5.6.

5.1 Login: Entrada no Sistema

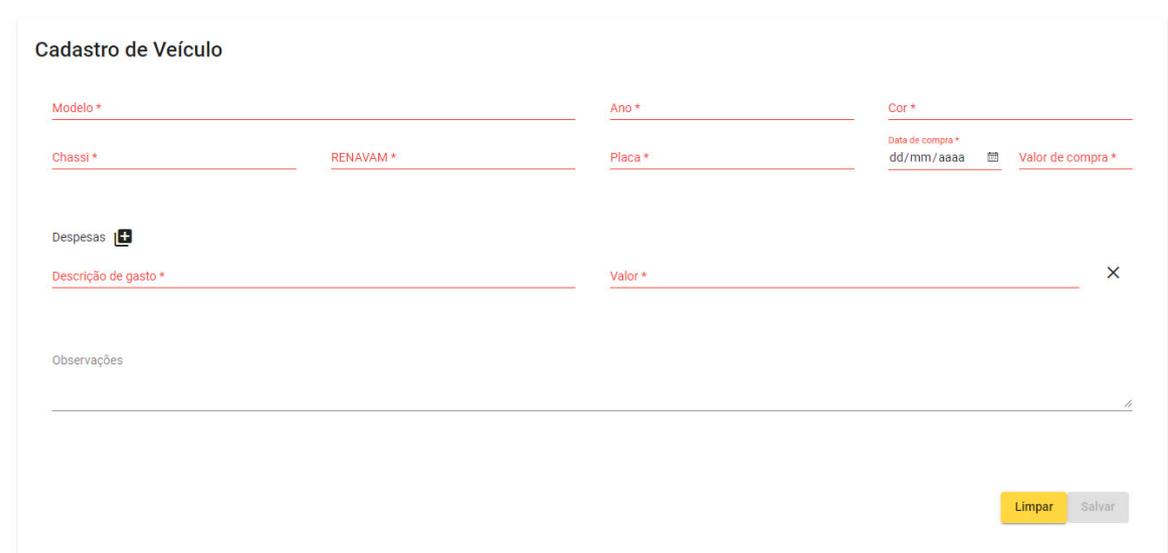
Ao entrar no sistema, a página inicial pede o *login* e a senha (Figura 8), caso o usuário falhe em fazer o *login* (Figura 26), o sistema exibe uma mensagem de erro. Se as entradas fornecidas estiverem corretas, o usuário é redirecionado para a página principal da aplicação, conforme mostra a Figura 9. A partir daí, o usuário pode selecionar pelo menu superior as ações que irá fazer. É possível fazer o *logout* do sistema a qualquer momento, seguindo o caminho de configurações (ícone da engrenagem) e então selecionar a opção “Sair”.

Figura 26 – Falha ao fazer o *login*.

5.2 Cadastrar

A partir do menu de cadastros, o usuário pode fazer a inserção de dados essenciais para a criação das operações, que serão apresentadas no texto mais a diante. As opções disponíveis são Veículo, Cliente, Funcionário e Usuário e são descritas a seguir:

- Veículo: o usuário deverá preencher todos os campos marcados com o asterisco (*), somente assim o botão de salvar fica disponível ao usuário. Como apresentado na Figura 27, se o usuário passar pelo campo sem preenchê-lo, ele se destacará em vermelho. O campo múltiplo “despesas” não é obrigatório, apesar de se iniciar com uma instância, podendo ser removido através do “X”. Caso o usuário opte por limpar os campos, deverá clicar no botão de limpar. Após o preenchimento de todos os campos e depois de salvar, o usuário será redirecionado a Lista de Veículos;



Cadastro de Veículo

Modelo * Ano * Cor *

Chassi * RENAVAL * Placa * Data de compra * Valor de compra *

dd/mm/aaaa

Despesas 

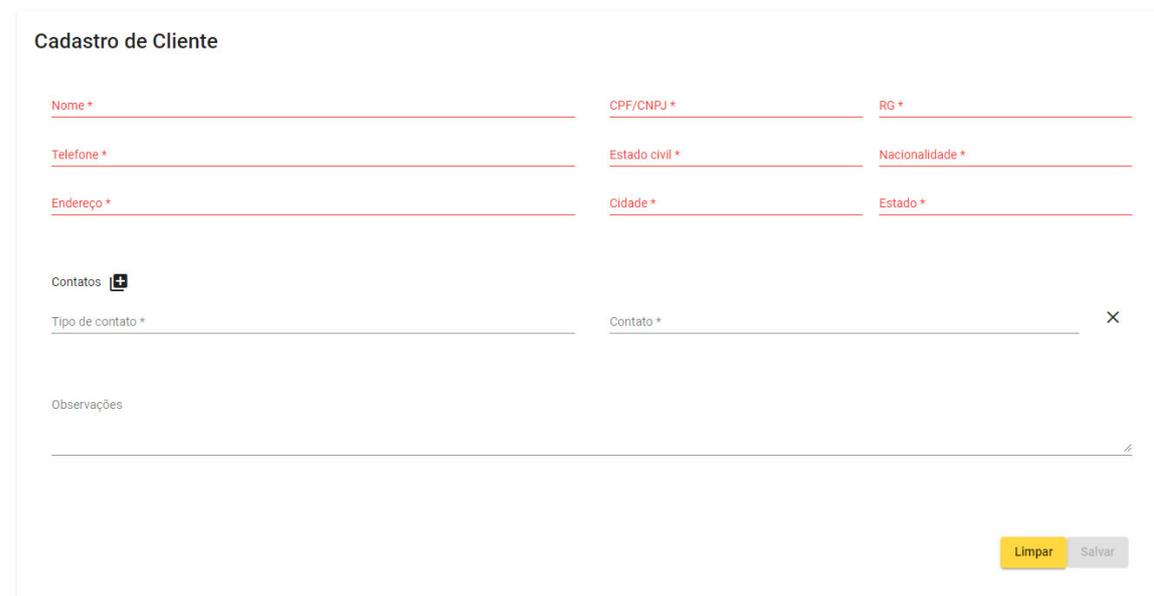
Descrição de gasto * Valor * 

Observações 

Figura 27 – Campos obrigatórios ao cadastrar um veículo.

- Cliente: para o cadastro de cliente, o usuário deverá preencher todos os campos obrigatórios (Figura 28), e assim o botão salvar ficará habilitado. Caso opte por limpar todos os campos, poderá fazê-lo em qualquer momento. Ao salvar, o usuário é redirecionado para a Lista de clientes;



Cadastro de Cliente

Nome * CPF/CNPJ * RG *

Telefone * Estado civil * Nacionalidade *

Endereço * Cidade * Estado *

Contatos 

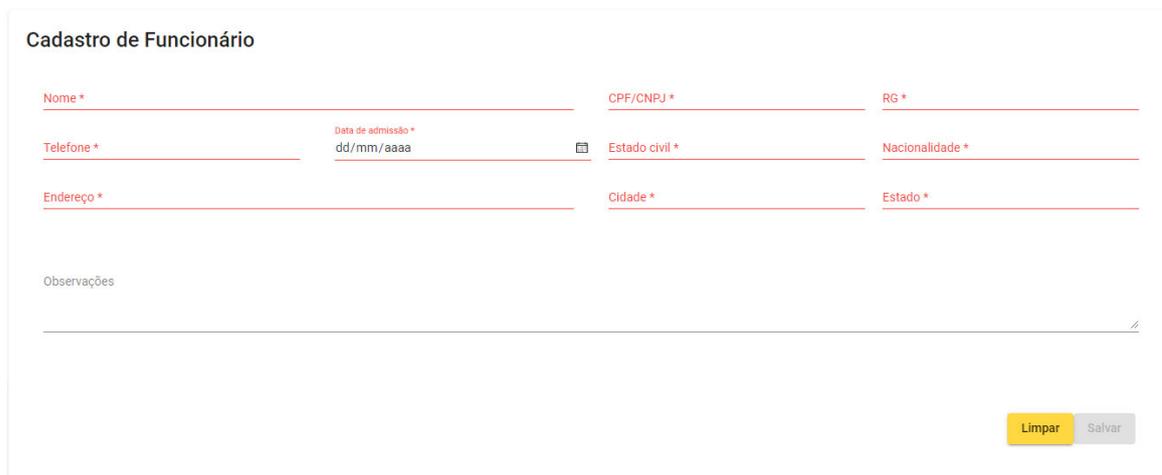
Tipo de contato * Contato * 

Observações 

Figura 28 – Campos obrigatórios ao cadastrar um cliente.

- Funcionário: após o preenchimento dos campos obrigatórios, o usuário poderá salvar seu cadastro no sistema. Na Figura 29, os campos obrigatórios são similares aos do Cadastro de cliente, exceto pela data de admissão. Há ainda a opção de limpar todos os campos nesse formulário. Após todos os campos serem preenchidos e depois de salvar o formulário, será redirecionado a Lista de Funcionários;



Cadastro de Funcionário

Nome * CPF/CNPJ * RG *

Telefone * Data de admissão * Estado civil * Nacionalidade *

dd/mm/aaaa

Endereço * Cidade * Estado *

Observações

Limpar Salvar

Figura 29 – Campos obrigatórios ao cadastrar um funcionário.

- Usuário: o cadastro de usuário (Figura 30) é feito somente pelo administrador e é obrigatório o preenchimento de usuário, senha, e permissões, esse último é um campo múltiplo. Após preencher os dados e, depois de salvar o cadastro, o usuário será redirecionado a lista de usuários.



Cadastro de usuário

Usuário * Senha *

Permissões * Ativo?

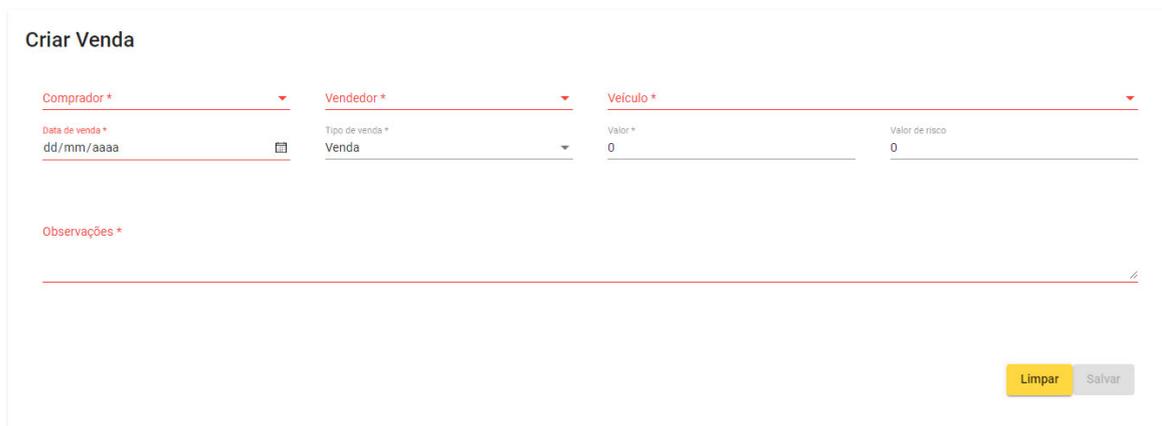
Limpar Salvar

Figura 30 – Campos obrigatórios ao cadastrar um usuário.

5.3 Operações

O módulo de operações consiste na criação das vendas e também de visualização de estatísticas geradas pela compra de veículos e vendas. A utilização da criação da venda, por exemplo, dependerá dos cadastros feitos dos clientes, funcionários e veículos, enquanto a visualização das estatísticas, das vendas feitas e dos veículos em estoque:

- Venda: deve-se preencher todos os campos da venda (Figura 31) para habilitar o botão de “salvar”. Importante mencionar que o campo veículo contém somente veículos não vendidos. Após a criação da venda, o usuário é redirecionado a Lista de Vendas;



O formulário 'Criar Venda' apresenta os seguintes campos obrigatórios:

- Comprador *
- Vendedor *
- Veículo *
- Data de venda * (formato dd/mm/aaaa)
- Tipo de venda * (menu suspenso com 'Venda' selecionado)
- Valor * (campo numérico com '0' preenchido)
- Valor de risco (campo numérico com '0' preenchido)
- Observações * (campo de texto)

Na base do formulário, há dois botões: 'Limpar' (em amarelo) e 'Salvar' (em cinza).

Figura 31 – Campos obrigatórios ao criar uma venda.

- Visualizar estatística: para visualizar estatísticas, é importante que já tenha sido cadastrados produtos e criadas as vendas. Conforme mostra a Figura 15, deve-se colocar uma data de início e uma data de fim para análise e clicar em “buscar”. Como padrão, o sistema traz o início e fim do mês atual, porém, qualquer intervalo pode ser usado.

5.4 Listas

As listas contêm todas as informações persistidas no sistema, o ponto de redirecionamento é feito quando se conclui um cadastro. As listas são do tipo expansível, então, os dados são exibidos por completo quando selecionado pelo usuário, mostrando somente uma prévia. Os itens contêm as opções de serem editados ou excluídos.

- Veículos: a lista de veículos (Figura 16) contêm cada veículo que foi cadastrado no sistema ordenado pela data de compra. Os dados principais apresentados são o modelo e a placa, para visualizar todos detalhes, deve-se clicar no item. Para editar um veículo, deve-se clicar no botão “Editar” e o usuário será redirecionado para a tela de cadastro, contendo todas as informações daquele item. Para excluir um veículo, deve-se clicar em excluir, e então a página será atualizada sem aquele item;
- Clientes: na lista de cliente (Figura 18) pode-se encontrar todos os clientes persistidos no sistema ordenados pelo nome. Os itens da lista exibem o nome e o CPF e, para visualizar todos detalhes, deve-se clicar em um deles. Para editar os dados do Cliente, deve-se clicar em “Editar” e o usuário será redirecionado para a tela de cadastro que contêm todos os campos do item. Ao clicar em excluir, a página será recarregada com todos itens, exceto aquele que foi excluído;
- Funcionários: similar a lista de clientes, a lista de funcionários (Figura 19) mostra o nome do funcionário em conjunto com seu CPF. Ao expandir o item, o usuário

poderá clicar em editar e será redirecionado para a tela de cadastro que contém todas as informações daquele item. Ao clicar em “Excluir” o item será removido e a página é atualizada;

- **Usuários:** a lista de usuários (Figura 17) apresenta todos usuários ativos e não ativos, eles são exibidos em ordem alfabética de *login*. Para editar, o usuário deve selecionar um item e clicar em “Editar”, em seguida, o usuário é redirecionado para a página de cadastro a qual contém todas informações, exceto a senha, a qual será redefinida. Para excluir, o usuário deve clicar em “Excluir”, então a página será atualizada e o item não será mais exibido;
- **Vendas:** conforme é apresentado pela Figura 20, o cabeçalho das vendas é o nome do cliente juntamente com o modelo do carro, sendo que os itens ordenados pela data de venda. Para editar uma venda, deve-se expandir o item e clicar em “Editar”, então será redirecionado para o cadastro de venda, com todos itens carregados. Ao Excluir uma venda, a página será recarregada com todos os outros itens. A opção de Gerar Contrato, representado pela prancheta, direciona o usuário para a tela de criação de contrato (Figura 21). Quando o contrato é tipo “troca”, o usuário deve preencher os dados do veículo cedido na troca. Se foi utilizado mais de um veículo na troca, deve-se adicionar mais um veículo utilizando o botão de adição. Se o usuário deseja exportar para PDF, deve-se clicar no botão da prancheta.

5.5 Configurações

Em configurações (Figura 32) o usuário pode ver detalhes da sua conta, além de trocar sua senha e fazer *logout*. Na hora de atualizar a senha, se a combinação de senhas não for válida (Figura 33), o sistema exibe um retorno mostrando que a combinação não é válida. Após inserir a configuração certa, assim como a senha atual, o sistema atualizará a senha corrente e retornará a página dos detalhes da conta.

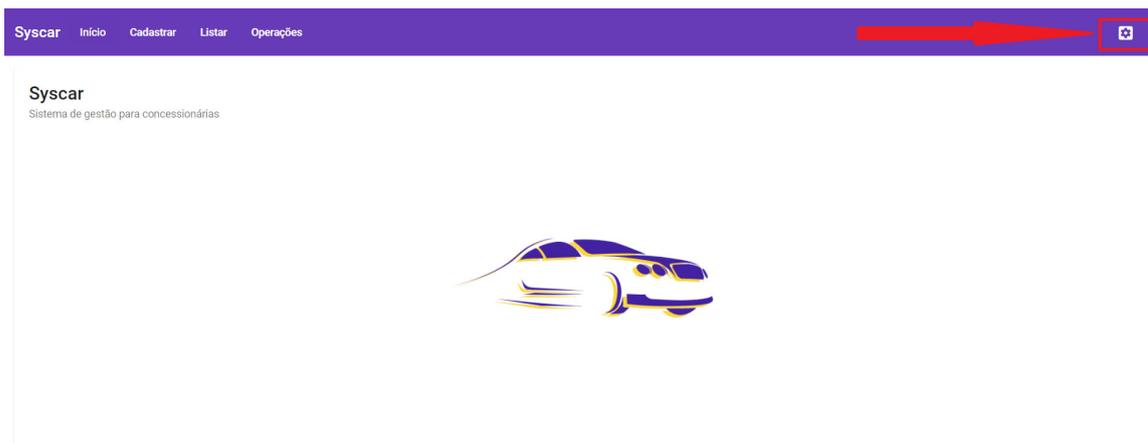


Figura 32 – Menu Configurações.

A screenshot of the 'Atualizar Senha' (Change Password) form. The form has three input fields: 'Senha atual' (Current password), 'Nova senha' (New password), and 'Confirmar senha' (Confirm password). Each field contains a series of dots representing masked text. A red error message is displayed below the 'Nova senha' field: 'Os campos devem ter a mesma senha' (Fields must have the same password). A button labeled 'Alterar senha' (Change password) is located at the bottom right of the form.

Figura 33 – Erro ao atualizar senha.

5.6 Considerações

Ao considerar que o ponto de partida são os requisitos funcionais descritos na Tabela 2, podemos observar que os requisitos RF-01 até o RF04 se fazem presentes no menu de cadastros, tendo listas e a partir das mesmas sendo possível editar e excluir um registro. Para o requisito RF-05, o cadastro está presente como uma operação, mas tendo sua própria lista com que além de excluir e deletar, oferece a possibilidade de gerar o contrato. Os requisitos RF-07 e RF-08 estão representados pela opção de visualizar estatísticas e o RF-09 - Alteração de senha é possível através do menu de configurações. O requisito RF-10 é mostrado quando o usuário acessa o sistema com a página de login e o Logout fica em configurações.

A divisão das funcionalidades entre módulos tem como foco a usabilidade do usuário no sistema, agrupando operações comuns como o cadastro, as listas contendo as opções

de editar e excluir e as operações do sistema em um só menu. Sendo assim dividido, os usuários terão mais facilidade ao se adaptar ao sistema, pois funcionalidades similares estão agrupadas.

As funcionalidades aqui apresentadas foram homologadas em conjunto com o usuário em duas reuniões distintas. A primeira reunião se homologou todos os CRUDs, cobrindo do RF-01 ao RF-05, além do *login* e do *logout*. O assunto da segunda reunião foi o RF-06 ao RF-08, demonstrando a geração do contrato e a visualização das estatísticas. Ambas reuniões foram feitas via ferramenta Team Viewer¹, onde o usuário operou diretamente o sistema, utilizando todas as funcionalidades mostradas. O escopo das reuniões era estabelecido e acordado previamente, assim o usuário teria ciência do que iria testar. Após a primeira e a segunda reunião, foram pedidas modificações e/ou adição em campos, mas nenhuma mudança no fluxo de tarefas.

¹ Team Viewer - <<https://www.teamviewer.com/pt-br/download/windows/>>

6 Conclusão

Neste trabalho foi desenvolvido o sistema Syscar, um sistema capaz de cobrir tarefas diárias de uma concessionária, como controle de estoque, gerenciamento de clientes e gestão de gastos. O lançamento da versão finaliza o escopo que foi proposto para esse projeto, tendo como o próximo passo a implementação e o uso pelos usuários finais.

A partir do modo de trabalho e das necessidades do usuário foram criados os requisitos funcionais e não funcionais para a construção do sistema. Através dos requisitos, foram gerados os diagramas que compõe esse trabalho, como o diagrama de casos de uso, que identificou os atores que vão utilizar o sistema. Através do diagrama de atividades, pode-se ver como funcionaria o fluxo de gerar um contrato e com o modelo ER, o que poderia se esperar do sistema em questão de entidades persistidas. A escolha das tecnologias para o desenvolvimento se deu com base no estudo e pesquisa de trabalhos relacionados além de soluções existentes no mercado, escolhendo-se duas plataformas muito usadas e com grandes comunidades: Angular e Spring. Com a escolha das tecnologias, foi possível traçar os requisitos mínimos para implementar o sistema em uma máquina local e dedicada.

O diagrama de sequência ilustrou como seria a interação do sistema usando como exemplo a criação de uma venda, mostrando que uma chamada partindo do *frontend* passa pelo *backend*, e o mesmo faz a requisição ao banco de dados, recuperando os dados ou persistindo-os. O desenvolvimento do *backend* e *frontend* gerou as funcionalidades do sistema, envolvendo todos os requisitos funcionais e não funcionais levantados no começo do trabalho. O *frontend* foi testado através de testes caixa preta, e devido a arquitetura de microsserviço, os testes do *backend* puderam ser realizados utilizando o *Postman*, assim certificando que todos métodos estão servindo às suas finalidades.

Como resultado tem-se um sistema capaz de persistir informações de clientes, veículos e funcionários, além de listar esses dados, com a opção de modificar e/ou removê-los. Com o Syscar também é possível verificar o balanço sazonal das vendas e do estoque cadastrado, além de cobrir tarefas de gestão de vendas e geração contratos de venda ou troca de veículos. Para o administrador, o sistema conta com as opções de gerenciar os usuários e suas permissões.

6.1 Limitações do trabalho

A disponibilização do sistema será feita em um servidor local dentro da dependências da empresa. Comparado com um serviço de nuvem, isso causa uma restrição de acesso ao sistema somente dentro rede interna, o que dificulta o acesso quando é necessário um

serviço *home office*, por exemplo. Outra limitação de um servidor físico local seria a falta de escalabilidade, que vai necessitar de manutenção para qualquer melhoria de configuração, além de manutenções preventivas e isso requer recursos humanos capacitados.

Outra limitação encontrada ao utilizar o sistema é a ausência de paginação e filtros de busca para as listas, conforme foi apresentado na Seção 5.4, o que dificulta a busca de itens ao longo do tempo. Um exemplo seria buscar uma venda que aconteceu há muito tempo, onde o usuário teria que utilizar a rolagem para buscar um determinado contrato em específico.

Em caso do usuário final utilizar um sistema de gestão financeira em seu fluxo de trabalho, por exemplo, o Syscar ainda não conta com formas de integração a esse sistema e nenhum outro. Outro ponto importante seria o fato das bibliotecas utilizadas no sistema ficarem obsoletas com o tempo, as quais necessitam de atualização e manutenção no código.

6.2 Contribuições

Comparando-se a maneira com que os colaboradores da empresa costumam trabalhar com planilhas de controle e geração de contratos digitados manualmente, o sistema oferece formas de se diminuir o trabalho manual repetitivo feito pelos colaboradores da empresa. O cadastro de clientes e veículos no sistema agiliza a consulta na hora de se criar uma venda nova, o que deixa para o sistema a responsabilidade de controlar o estoque de venda. Além disso, a criação de contratos a partir da venda substitui o preenchimento manual que antes eram feitos a partir de documentos no Microsoft Word, o que diminui o trabalho do usuário ao se criar um contrato. A visualização de estoque e vendas de forma sazonal é independente da criação de filtros e macros no Microsoft Excel e automatizada completamente no produto.

Diante disso, a importância deste projeto é entregar um produto final ao usuário que irá automatizar todos os serviços de processos internos realizados no dia a dia da empresa, o que traz agilidade na realização de tarefas tanto para os usuários da empresa bem como para os clientes da empresa. Adicionalmente, um sistema automatizado facilita a gestão do modelo de negócios da empresa e, certamente, será uma ferramenta importante para tomada de decisões e é um primeiro passo para realização de melhorias contínuas no modelo de negócio da empresa.

6.3 Trabalhos Futuros

Como melhoria para a gestão de clientes, propõe-se a criação de uma funcionalidade de vendas futuras. Imaginando um cenário em que um cliente chegue até a loja mas não

tem o veículo que ele deseja, assim pode ser cadastrado no sistema o desejo desse cliente por um veículo, uma vez que o veículo fosse adquirido, o sistema exibiria um alerta sobre esse cadastro de venda futura.

Visando reduzir as limitações desse trabalho, a criação de filtros e paginação para as listas um impacto positivo ao utilizar o sistema, como redução de tráfego de rede, além de a resposta de navegação pelas listas. Isso também pode trazer maior escalabilidade ao sistema, uma vez que poderia ser utilizado para consultar vendas mais antigas de maneira mais flexível.

Como um possível aprimoramento do produto, sugere-se a adaptação do mesmo aos serviços disponibilizados em nuvem. Isso pode ser alcançado com a criação de uma nova *branch* a partir da *master*, voltada para configurar todo o software do sistema com os serviços disponibilizados em nuvem. A utilização do serviço em nuvem traria vantagens como mobilidade no acesso e escalabilidade, reduzindo os custos da necessidade de manutenções, mas aumentando com o valor dos custos do armazenamento.

Referências

- ANGULAR. *Angular - Component*. 2020. Último acesso em 7 de dezembro de 2020. Disponível em: <<https://angular.io/api/core/Component>>. Citado na página 39.
- BASTOS, R.; JAQUES, P. Antares: Um sistema web de consulta de rotas de ônibus como serviço público. *Revista Brasileira de Computação Aplicada*, v. 2, p. 41–56, 01 2010. Citado na página 25.
- BOEDER, J.; GROENE, B. *The Architecture of SAP ERP: Understand how successful software works*. [S.l.]: tredition, 2014. ISBN 9783849576622. Citado na página 20.
- CUNHA, A. et al. Sistema computacional web para controle de gestão da produção de café. *Coffe Science*, v. 6, p. 344–354, 06 2017. Citado na página 24.
- ELEUTERIO, M. *Sistemas De Informações Gerenciais Na Atualidade*. [S.l.]: INTERSABERES, 2015. ISBN 9788544302859. Citado 2 vezes nas páginas 15 e 18.
- ELMASRI, R.; NAVATHE, S. *Sistemas de banco de dados*. [S.l.]: PEARSON BRASIL, 2011. ISBN 9788579360855. Citado 2 vezes nas páginas 31 e 34.
- ESTÁCIO, M. et al. Desenvolvimento de um sistema web de gerenciamento de conteúdo comercial autônomo. *Colloquium Exactarum*, v. 6, p. 55–70, 08 2014. Citado na página 23.
- FATALA, A. *André Fatala e a transformação que multiplicou o Magalu por 700*. 2020. Disponível em: <<https://www.infomoney.com.br/stock-pickers/andre-fatala-e-a-transformacao-que-multiplicou-o-magalu-por-700/>>. Citado na página 20.
- FGV. *Pesquisa Anual do Uso de TI*. 2020. Último acesso em 26 de novembro de 2020. Disponível em: <<https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti>>. Citado 2 vezes nas páginas 15 e 21.
- FILHO, M. P. d. C. Os sistemas de informação e as modernas tendências da tecnologia e dos negócios. *Revista de Administração de Empresas*, SCIELO, v. 34, p. 33 – 45, 12 1994. ISSN 0034-7590. Citado na página 19.
- GAMMA, E. et al. *Padrões de Projetos: Soluções Reutilizáveis*. [S.l.]: Bookman, 2006. ISBN 9788573076103. Citado na página 37.
- JÚNIOR, J. W. d. S. *GarageParts: Sistema de Controle de Consumo, Deslocamento e Manutenções de Veículos Pessoais*. 2019. Último acesso em 16 de dezembro de 2020. Disponível em: <<https://monografias.ufop.br/handle/35400000/2482?locale=en>>. Citado na página 26.
- LAUDON, K.; LAUDON, J. *Sistemas de informação gerenciais*. [S.l.]: PEARSON BRASIL, 2013. v. 11. ISBN 9788543005850. Citado 4 vezes nas páginas 15, 18, 19 e 20.
- MARTIN, R. *Arquitetura Limpa: O guia do artesão para estrutura e design de software*. [S.l.]: Alta Books, 2019. (Robert C. Martin). ISBN 9788550808161. Citado na página 41.

MOHAPATRA, S. *Business Process Automation*. [S.l.]: PHI Learning, 2009. ISBN 9788120339279. Citado 2 vezes nas páginas 15 e 19.

NIST. *Secure Hash Standard*. 2002. Último acesso em 16 de dezembro de 2020. Disponível em: <<https://csrc.nist.gov/csrc/media/publications/fips/180/2/archive/2002-08-01/documents/fips180-2.pdf>>. Citado na página 38.

ORACLE. *Core J2EE Patterns - Data Access Object*. 2002. Último acesso em 7 de dezembro de 2020. Disponível em: <<https://www.oracle.com/java/technologies/dataaccessobject.html>>. Citado na página 37.

SOMMERVILLE, I. *Engenharia de software*. [S.l.]: PEARSON BRASIL, 2011. ISBN 9788579361081. Citado 5 vezes nas páginas 29, 31, 32, 33 e 40.

TOTVS. *O que é ERP? Guia completo sobre o assunto - TOTVS*. 2019. Último acesso em 26 de novembro de 2020. Disponível em: <https://www.totvs.com/blog/erp/o-que-e-erp/#por_que_escolher_um_erp_totvs>. Citado na página 21.