

UNIVERSIDADE FEDERAL DO PARANÁ

GIULIA GUARISE GUTIERREZ
RAPHAEL PEREIRA RODRIGUES

MARVIN: SISTEMA DE GERENCIAMENTO PARA CENTROS AUTOMOTIVOS

CURITIBA

2022

GIULIA GUARISE GUTIERREZ
RAPHAEL PEREIRA RODRIGUES

MARVIN: SISTEMA DE GERENCIAMENTO PARA CENTROS AUTOMOTIVOS

Trabalho de conclusão de curso apresentado ao curso de Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. João Eugenio Marynowski

CURITIBA

2022



UNIVERSIDADE FEDERAL DO PARANÁ

ATA DE REUNIÃO

TERMO DE APROVAÇÃO

Giulia Guarise Gutierrez

Raphael Pereira Rodrigues

MARVIN: SISTEMA DE GERENCIAMENTO PARA CENTROS AUTOMOTIVOS

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. João Eugenio Marynowski

Orientador – SEPT/UFPR

Prof. Mário de Paula Soares Filho

SEPT/UFPR

Prof. Rafael Romualdo Wandresen

SEPT/UFPR

Curitiba, 12 de maio de 2022.

Documento assinado eletronicamente por **JOAO EUGENIO MARYNOWSKI, PROFESSOR DO MAGISTERIO SUPERIOR**, em 12/05/2022, às 21:11, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **MARIO DE PAULA SOARES FILHO, PROFESSOR ENSINO BAS/TEC/TECNOL**, em 12/05/2022, às 22:09, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **RAFAEL ROMUALDO WANDRESEN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 12/05/2022, às 22:14, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **4508332** e o código CRC **8BB3E689**.

AGRADECIMENTOS

Agradecemos ao cliente desse projeto, o senhor Marco Aurélio Rodrigues, por confiar em nós com esse sistema tão importante para sua empresa, e acompanhar o seu desenvolvimento ao longo do tempo.

Agradecemos aos nossos familiares e amigos pelo apoio e incentivo durante todo o processo, nem temos palavras pra descrever o quanto importante eles são em nossas vidas. Temos muita sorte de poder contar com a paciência e carinho deles todos os dias.

Agradecemos ao nosso orientador Prof. Dr. João Eugênio Marynowski por nos apoiar e aconselhar em tantos momentos. O Marvin não seria o que é se não fosse por ele.

Agradecemos à banca, composta pelos Professores Dr. Mario de Paula Soares Filho e Dr. Rafael Romualdo Wandresen, por aceitarem ser os avaliadores do presente trabalho.

E a todos que direta ou indiretamente fizeram parte da nossa formação, o nosso muito obrigado.

RESUMO

Um centro automotivo é um estabelecimento responsável pela manutenção e melhorias de automóveis e, assim como empresas de diferentes ramos, pode se beneficiar com a digitalização e automatização de processos através do uso de softwares. O proprietário da Automar, um centro automotivo em Curitiba, buscava essa melhoria para melhor atender seus clientes e agilizar suas atividades diárias, concomitantemente com possuir um sistema que acompanhasse o futuro crescimento de sua empresa. Com esse objetivo o Marvin foi criado, um sistema gerenciador que automatiza os principais processos da empresa, utilizando uma arquitetura de microsserviços e encapsulamento de aplicações em contêineres para assegurar sua evolução em paralelo com a empresa. Desenvolvido utilizando a metodologia Scrum adaptada para melhor se encaixar na rotina da equipe e cliente externo, o sistema entrega robustez e escalabilidade em interfaces descomplicadas e adaptáveis.

Palavras-chave: Automatização. Centro Automotivo. Microserviço. Container.

ABSTRACT

An automotive center is a business responsible for the maintenance and improvements of automobiles and, like other companies in different fields, can benefit from the digitalization and automation of business processes through the use of software. The owner of Automar, an auto center in Curitiba, wanted that upgrade to better assist his customers and to speed up daily tasks, as well as to have a future-proof software. With that in mind Marvin was created, a management software that automates the main company's activities and utilizes a microservices architecture with application encapsulation in containers to secure its evolution in parallel with the company's. Developed using an adapted Scrum methodology to better fit the group and external client's routine, the software delivers sturdiness and scalability with uncomplicated and adaptable interfaces.

Keywords: Automation. Automotive Center. Microservice. Container.

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVO GERAL	9
1.2 OBJETIVOS ESPECÍFICOS	10
1.3 JUSTIFICATIVA	10
1.4 ORGANIZAÇÃO DO TRABALHO	10
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 FUNCIONAMENTO DO AUTO CENTER AUTOMAR	11
2.2 SISTEMAS SIMILARES	12
2.2.1 Car Online 365	12
2.2.2 Oficina Integrada	13
2.2.3 Oficina Inteligente.....	14
2.2.4 UltracarWeb	15
2.3 ANÁLISE DE SOFTWARES SIMILARES.....	17
3 MATERIAIS E MÉTODOS	18
3.1 MATERIAIS	18
3.1.1 Padrões de Projeto.....	18
3.1.2 Back-end	19
3.1.3 Front-end.....	20
3.1.4 Infraestrutura	20
3.1.5 Demais Ferramentas	21
3.2 DESENVOLVIMENTO ÁGIL DE SOFTWARE	21
3.2.1 Adaptação do modelo Scrum	22
3.2.2 Desenvolvimento do Projeto Marvin.....	22
4 APRESENTAÇÃO DO SISTEMA	25
4.1 ARQUITETURA.....	25
4.2 FUNCIONAMENTO	29
4.2.1 Menu	29
4.2.2 Telas de Cliente, Orçamento e Agendamento.....	30
4.2.3 Telas de Cadastros	31
4.2.4 Confirmação de Mudança de Tela	32
4.2.5 Fechar Caixas de Diálogo (Tela Suspensa)	32
4.2.6 Estados Internos dos Componentes	32

4.3 TELAS DO SISTEMA	33
4.3.1 Tela de Início.....	34
4.3.2 Tela de Cadastro de Serviços	35
4.3.3 Tela de Cadastro de Fornecedores	36
4.3.4 Tela de Cadastro de Estoque.....	36
4.3.5 Telas de Listagem	37
4.3.6 Tela de Cliente	38
4.3.7 Tela de Orçamento.....	40
4.3.8 Tela de Agendamento	43
5 CONSIDERAÇÕES FINAIS	44
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS	44
REFERÊNCIAS.....	46
APÊNDICE 1 – LISTA DE REQUISITOS FUNCIONAIS	47
APÊNDICE 2 – DIAGRAMA DE CASOS DE USO	48
APÊNDICE 3 – LISTA DE ÍCONES DAS TELAS.....	49
APÊNDICE 4 – TELAS GENÉRICAS DO SISTEMA.....	50
APÊNDICE 6 – DIAGRAMA DE CLASSES	88
APÊNDICE 7 – DIAGRAMAS DE SEQUÊNCIA.....	89
APÊNDICE 8 – DIAGRAMAS DE SEQUÊNCIA DE IMPLEMENTAÇÃO	94
APÊNDICE 9 – DIAGRAMA RELACIONAL.....	98
APÊNDICE 10 – DESCRIÇÃO DOS SPRINTS DA FASE DE MODELAGEM	99
APÊNDICE 11 – DESCRIÇÃO DOS SPRINTS DA FASE DE CONSTRUÇÃO	101
APÊNDICE 12 – GUIA DE INSTALAÇÃO DO SISTEMA MARVIN	104

1 INTRODUÇÃO

Auto center é um local que oferece a solução de problemas automobilísticos por profissionais de alto nível de conhecimento técnico, e oferta serviços mais completos que os encontrados em oficinas mecânicas, como soluções elétricas e de borracharia (ULTRACAR, 2019).

Dada a natureza de pequeno porte de algumas empresas e a quantidade limitada de recursos que muitas vezes possuem, é necessário que o processo de agendamento e cadastro de clientes, fornecedores e orçamentos sejam realizados em forma digital para que estas empresas iniciem seu funcionamento de forma organizada, consistente e sem custos mensais.

Gai (2020) descreve que é possível observar nos últimos anos uma rápida transição de aplicações e servidores monolíticos para sistemas virtualizados e distribuídos. Os sistemas distribuídos possuem maior escalabilidade e flexibilidade, podendo assim escalar de forma rápida conforme a demanda e o aumento da empresa, mas em contrapartida possuem um alto valor de investimento para seu desenvolvimento.

Deste modo, alguns centros automotivos ficam impossibilitados de terem acesso a um sistema distribuído que seja customizável e escalável conforme suas demandas, restando para esses o uso de sistemas monolíticos que muitas vezes são de difícil integração e escalabilidade. Isso faz com que a empresa tenha maiores gastos com a contratação de funcionários que atuem como intermediários da comunicação entre os sistemas e de servidores que ficarão em maior parte do tempo ociosos para suprir uma possível demanda.

1.1 OBJETIVO GERAL

O presente trabalho tem o objetivo de levantar os requisitos específicos da empresa Automar através da metodologia Scrum e desenvolver um sistema distribuído que permita o gerenciamento de atendimento de clientes de um centro automotivo.

1.2 OBJETIVOS ESPECÍFICOS

- Possibilidade de escalonamento vertical e horizontal do sistema;
- Possibilidade de acesso do sistema através de diferentes dispositivos;
- Facilidade de integração com sistemas externos e possíveis sistemas futuros da empresa;
- Interfaces de usuário facilitadas e com curva de aprendizado baixa;

1.3 JUSTIFICATIVA

O presente sistema foi desenvolvido com base nas demandas do centro automotivo Automar, sendo que o levantamento de requisitos foi feito diretamente com o proprietário da empresa para melhor compreender as especificidades do negócio e atender de forma mais abrangente as necessidades da organização.

Dessas necessidades as que se destacam são as de tornar o sistema simples para o uso diário e contínuo, otimizando o atendimento ao cliente sem complicá-lo, e a de possibilitar que a empresa cresça sem abandonar o software já implantado.

A empresa atualmente faz todos os processos levantados de forma manual, diretamente em papel, o que não é ideal tendo em vista o possível extravio de documentos e a desorganização sistêmica que esses processos manuais podem acarretar no futuro.

Deste modo, o presente trabalho propõe o Marvin: um sistema composto por microsserviços encapsulados em contêineres, com o propósito de atender o cerne de centros automotivos que necessitem da automatização descomplicada do processo de atendimento de clientes, de forma escalável, flexível e pronto para ser implantado em nuvem, facilitando deste modo o gerenciamento dos dados da empresa e o escalonamento do sistema em conjunto com a empresa em questão.

1.4 ORGANIZAÇÃO DO TRABALHO

No Capítulo 1 apresenta-se o contexto, o cliente e suas necessidades, os objetivos gerais e específicos, e as justificativas que englobam este trabalho. No Capítulo 2 fundamenta-se o negócio, bem como compara o sistema proposto com softwares similares. No Capítulo 3 explica-se sobre os materiais e métodos utilizados

e como o desenvolvimento ocorreu. No Capítulo 4 apresenta-se a arquitetura do sistema, suas interfaces e funcionamento. No Capítulo 5 tem-se as considerações finais e as recomendações para trabalhos futuros. Os apêndices contêm materiais extras que auxiliam no entendimento da proposta e na instalação do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado, nas respectivas seções, o funcionamento básico do centro automotivo Automar, nosso estudo de caso, seguido da apresentação dos sistemas disponíveis no mercado similares ao Marvin, o objeto de desenvolvimento deste trabalho.

2.1 FUNCIONAMENTO DO AUTO CENTER AUTOMAR

Como informado pelo dono da empresa Automar, os processos que envolvem sua rotina empresarial se dividem em etapas. A primeira é a procura do cliente pelos serviços, ocorrendo então a elaboração de um orçamento. Esse será formulado de acordo com as necessidades identificadas no automóvel, e dependerá de peças, produtos ou serviços que darão o custo total.

Os valores das peças e produtos vêm da pesquisa do funcionário que está realizando o atendimento do cliente com fornecedores diversos, visto que a empresa opta por não comprar itens antes da realização do serviço, com exceções para produtos utilizados por mais de uma marca e modelo de carro.

As próximas etapas se tornam opcionais, pois nascem da escolha do cliente em realizar o trabalho. Se estiver de acordo com o orçamento proposto, esse entra em contato para agendar uma data para realização do serviço. Após, leva o veículo no dia combinado, no qual os detalhes do orçamento são verificados e a ordem de serviço é assinada.

A ordem de serviço é um documento que oficializa o serviço a ser prestado, constando detalhes como custo de materiais e mão de obra necessária, sendo indispensável para a execução do trabalho (NADER, 2020).

O proprietário do centro automotivo Automar enfatiza que a ordem de serviço trata sobre o agendamento, a execução do orçamento previamente levantado, e a

autorização explícita do cliente de forma que esse consente com os serviços, peças, produtos e custo que constam no orçamento inicial.

Por fim, com a ordem de serviço acordada entre as partes, o trabalho especificado é realizado e o freguês efetua o pagamento.

O local que a empresa opera permite o atendimento de vários veículos simultaneamente, e por consequência os horários desses podem se sobrepor. Por essa razão, é imprescindível haver uma listagem acurada de agendamentos para o dia, bem como de suas respectivas ordens de serviço, que garantem a execução exata de trabalhos.

2.2 SISTEMAS SIMILARES

Com o propósito de fundamentar e compreender o escopo do problema de negócio que o sistema Marvin busca solucionar, foi feito uma pesquisa de softwares correlatos que abrangem as funções necessárias para atender a demanda de informatização de centros automotivos. É válido salientar que nenhum dos sistemas analisados são de código aberto, deste modo não houve acesso direto a dois dos quatro sistemas analisados.

Assim, esta seção se encarrega de apresentar os sistemas em questão e expõe uma análise das funcionalidades dos sistemas, quando existe a possibilidade de acesso direto, e caso impossibilitado de interagir com o software a análise é feita através das informações disponibilizadas pelas empresas desenvolvedoras através de seus respectivos sites.

2.2.1 Car Online 365

Desenvolvido pela empresa BitStorm¹, o software promete auxiliar nas rotinas da empresa e informatizar uma grande parte das responsabilidades em uma oficina. O sistema é separado em diferentes módulos especializados em diversos assuntos, e possui funcionalidades como orçamentos, agendamentos, ordens de serviços, e cadastro de clientes, veículos e fornecedores.

¹ CAR ONLINE 365. Disponível em: <<https://www.caronline365.com.br/v2/>>. Acesso em: 12 nov. 2021.

O desenvolvedor disponibiliza o software diretamente para download em seu próprio site, porém não foi possível executar a aplicação mesmo após inúmeras tentativas de instalação. A empresa também não disponibiliza diretamente no site os preços do software, mas providencia uma página para cotação.

FIGURA 1 – TELA DE CONTROLE DE PONTO DO SISTEMA CAR ONLINE 365



FONTE: SITE DO SOFTWARE CAR ONLINE 365 (2021)

2.2.2 Oficina Integrada

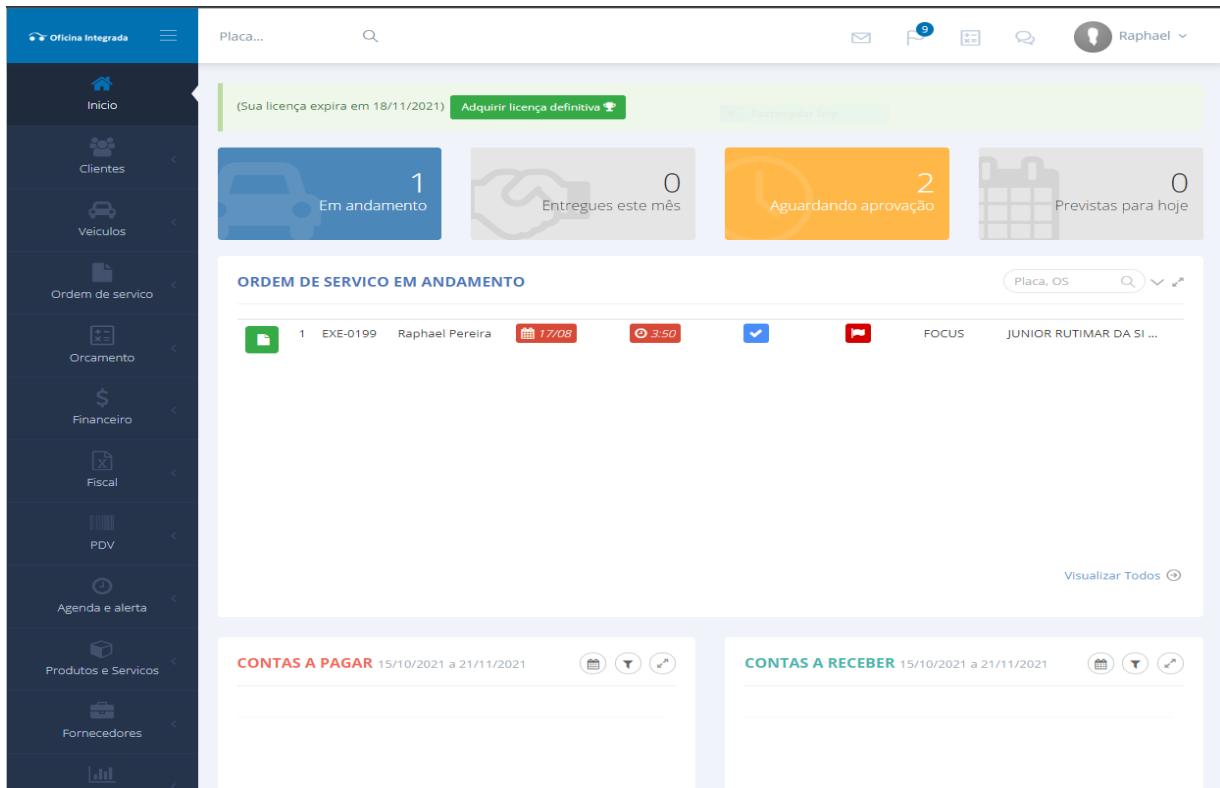
Desenvolvido pela empresa Mundomidia Ltda.², se trata de uma aplicação web distribuída através de planos de assinatura que variam de R\$ 59,90 (cinquenta e nove reais e noventa centavos) a R\$ 174,00 (cento e setenta e quatro reais) por mês, a depender da quantidade de usuários que utilizarão a aplicação. Esse é um dos sistemas em que foi possível o acesso direto através de login de teste disponibilizado pela própria empresa.

Fundamentado em testes próprios, é possível afirmar que o sistema possui navegação intuitiva e grande quantidade de funcionalidades pertinentes ao funcionamento diário de oficinas mecânicas, sendo o único com todas as que o Marvin

² OFICINA INTEGRADA. Disponível em: <<https://www.oficinaintegrada.com.br/software-gerencimento-oficina-mecanica/programa-gestao-oficina-mecanica-integrada/default.asp>>. Acesso em: 12 nov. 2021.

almeja: orçamento, agendamento, ordem de serviço, e cadastro de clientes, veículos, serviços, produtos e fornecedores. Além disso, têm algumas funcionalidades que permitem o cadastro com campos vazios, como a de veículo, que deixa que um automóvel sem nenhum campo preenchido seja salvo e vinculado a um cliente. A empresa não retornou contato após relato dos problemas identificados.

FIGURA 2 – TELA INICIAL DO SISTEMA OFICINA INTEGRADA



FONTE: SISTEMA OFICINA INTEGRADA (2021)

2.2.3 Oficina Inteligente

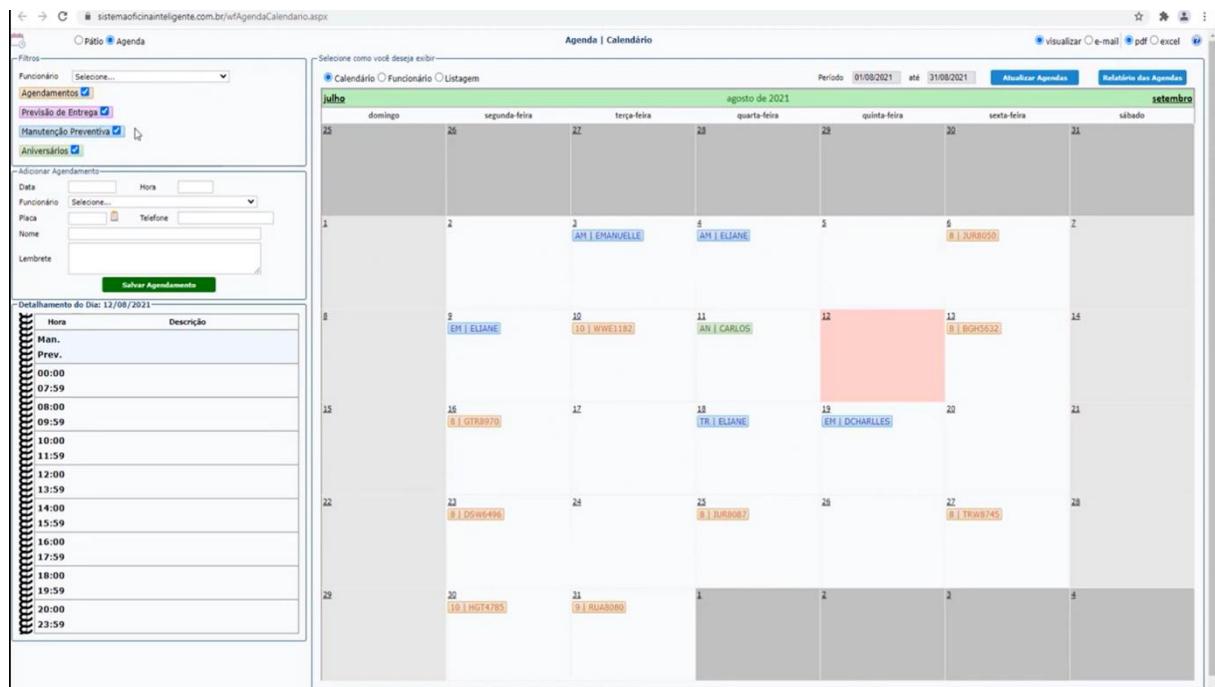
O sistema Oficina Inteligente, desenvolvido pela empresa homônima³, busca apresentar também uma solução de aplicação web. Não foi possível o acesso direto ao sistema, porém a desenvolvedora disponibiliza vários vídeos demonstrativos dos diferentes recursos que a aplicação oferece, como controle de pátio, emissão de nota fiscal, controle financeiro e outros.

³ OFICINA INTELIGENTE. Disponível em: <<https://oficinainteligente.com.br/>>. Acesso em: 13 nov. 2021.

A organização oferece planos que variam de R\$ 289 (duzentos e oitenta e nove reais) a R\$ 499 (quatrocentos e noventa e nove reais) ao mês, com a quantidade de recursos do sistema variando de acordo com o plano selecionado.

Apesar da aplicação cobrir uma gama de recursos necessários para o funcionamento de um auto center – as funcionalidades apresentadas no Oficina Integrada, exceto o cadastro de fornecedores – o alto preço mensal para utilização da aplicação pode tornar inviável sua utilização em centros automotivos de pequeno porte, onde uma grande parte dos recursos disponibilizados, mesmo na versão mais barata, possivelmente não serão utilizados.

FIGURA 3 – TELA DE AGENDA DO SOFTWARE OFICINA INTELIGENTE



FONTE: SITE DO SOFTWARE OFICINA INTELIGENTE (2021)

2.2.4 UltracarWeb

Desenvolvido pela empresa Ultracar Comércio em Informática Ltda., o UltracarWeb é uma aplicação web com acesso pela nuvem que promete simplificar e facilitar a gestão de uma oficina⁴.

Os valores da aplicação foram informados por telefone. Existe um valor inicial de R\$ 600 (seiscientos reais) para implantação e treinamento de seis horas com um

⁴ ULTRACARWEB. Disponível em: <<https://ultracarweb.com/>>. Acesso em: 22 nov. 2021.

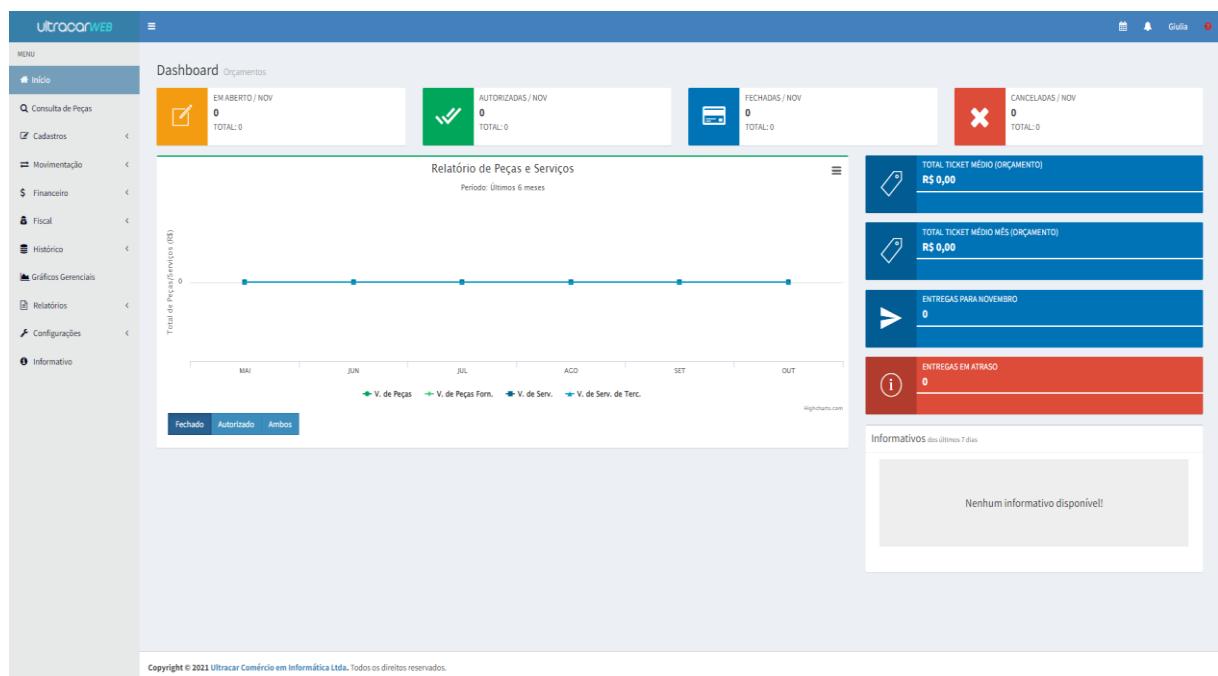
especialista do software. O custo do plano mensal para um único usuário é de R\$ 200 (duzentos reais), sendo acrescentado R\$ 30 (trinta reais) para cada novo acesso disponibilizado.

A empresa disponibiliza um teste de 10 dias para o sistema após breve cadastro no site, que foi o utilizado para testes próprios. Baseado nele, foi possível atestar que o software possui as funcionalidades de orçamento, ordem de serviço, cadastro de clientes, de veículos, serviços, produtos e fornecedores, e também recursos diferenciados como gráficos gerenciais e relatórios de produtividade, o que o destaca na área gerencial.

Entretanto, não possui opção de agendamento de serviço e frustra com mensagens de erro que dificultam a finalização de cadastros, como sinalizar para informar “Funcionário” com a identificação do campo sendo “Orçamentista”.

Outro detalhe percebido ao depurar a aplicação após demora na persistência de dados foi a grande quantidade de declarações *debug (debugger)* encontradas na aplicação, o que resulta em parada de execução caso haja alguma funcionalidade de depuração disponível (MDN WEB DOCS, 2021).

FIGURA 4 – TELA DE INÍCIO DO SOFTWARE ULTRACARWEB



FONTE: SISTEMA ULTRACARWEB (2021)

2.3 ANÁLISE DE SOFTWARES SIMILARES

Uma grande semelhança foi identificada entre os softwares analisados em relação as suas funcionalidades. Esta análise também facilitou a melhor compreensão do escopo do problema e de quais recursos compõem o centro destas aplicações.

As funcionalidades centrais selecionadas foram: cadastro de cliente, que guarda informações como nome e contato do freguês; cadastro de veículos, que permite que um carro seja registrado no sistema; orçamento, que registra informações como custo e serviços a serem efetuados para o cliente; agendamento, que possibilita reservar um horário para um atendimento; ordem de serviço, que oficializa o atendimento a ser prestado; cadastro de serviços, que armazena detalhes do mesmo; cadastro de produtos, que guarda dados de um item; e cadastro de fornecedores, que armazena informações de empresas que vendem objetos de interesse para o centro automotivo.

Foi possível também perceber diferenças em contextos que parecem iguais em primeiro plano, mas quando analisados com mais atenção é notável que eles possuem objetivos diferentes, como é o caso de orçamento e ordem de serviço.

TABELA 1 – COMPARATIVO DE SOFTWARES SIMILARES

Características	Aplicações			
	Car Online 365	Oficina Integrada	Oficina Inteligente	UltracarWeb
Plataforma Web		x	x	x
Cadastro de Clientes	x	x	x	x
Cadastro de Veículos	x	x	x	x
Orçamento	x	x	x	x
Agendamento	x	x	x	
Ordem de Serviço	x	x	x	x
Cadastro de Serviços		x	x	x
Cadastro de Produtos		x	x	x
Cadastro de Fornecedores	x	x		x
Software Livre				

FONTE: Os autores (2021)

A Tabela 1 apresenta as funcionalidades centrais de cada sistema, conforme citado nesta e na seção anterior. Também aponta se o sistema é em Plataforma Web – o sistema é acessado em um ambiente online – e se possui Software Livre, ou seja, se é gratuito e pode ser modificado se necessário.

Com base nisso, é perceptível que todos têm ótimas funcionalidades disponíveis, porém nenhum deles possui código aberto, desta forma restringindo o quanto customizada a aplicação ficaria com as necessidades da empresa Automar.

Outro ponto relevante para a empresa é uma separação bem definida entre produtos e peças. O primeiro se refere a elementos que podem ser armazenados em estoque e têm um fornecedor associado, enquanto o segundo representa itens específicos ou caros demais para serem estocados, e, portanto, só estarão relacionados a um orçamento.

Dentre os softwares similares encontrados, nenhum apresentou uma solução facilitada para esse problema.

3 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados e fundamentados as tecnologias, metodologias e conceitos utilizados na construção do Marvin, que é o sistema proposto neste trabalho.

3.1 MATERIAIS

Esta seção é responsável pelo detalhamento dos materiais utilizados no desenvolvimento do presente trabalho. Ela está separada por padrões de projeto, back-end, front-end e as ferramentas utilizadas para prover a infraestrutura.

3.1.1 Padrões de Projeto

Com o propósito de ser um sistema de fácil integração, implantação, alta escalabilidade e baixo acoplamento, o Marvin foi construído baseado em design orientado a domínio (DDD ou *Domain-Driven Design*) em concomitância com o princípio CQS (Separação de comandos e consultas ou *Command-Query Separation*) através da implementação de microsserviços. Todos estes microsserviços são encapsulados em contêineres Docker com o propósito de facilitar a implantação do

sistema e sua escalabilidade, visando com que seja possível atender de forma mais ágil as necessidades do cliente e com o propósito de que o sistema esteja pronto para que seja integrado com as práticas e ferramentas de DevOps utilizadas em processos de integração e entrega contínua (CI/CD).

Evans (2003) define o domínio da aplicação como a atividade ou interesse do usuário na qual o programa se aplica, ou seja, na realidade do contexto dos negócios de forma que a aplicação se conforme com o domínio do negócio e não o contrário.

No padrão CQS, como estabelecido por Bertrand (1997), apenas comandos devem modificar o estado de uma entidade, enquanto consultas são responsáveis apenas pela verificação de estado das entidades sem fazer nenhuma alteração nessas. Esse padrão facilita a aplicação do princípio da responsabilidade única, definido por Martin no livro Agile Software Development (2003, p.95), que desacopla as responsabilidades de comandos e consultas e facilita o escalonamento de uma aplicação de modo que comandos e consultas possam ser feitos em bancos e tecnologias diferentes, sem que haja grande impacto na implementação já feita da aplicação.

O Marvin utiliza também o padrão Mediator, que é definido por Gamma et al (1997) como um objeto que encapsula o modo que uma gama de objetos interage, o que facilita o desacoplamento através da inclusão de uma classe mediadora que fica responsável pela intermediação entre classes, de modo que essas não precisem mais se referenciar diretamente.

3.1.2 Back-end

Para o desenvolvimento do back-end foi utilizada IDE gratuita Visual Studio Community⁵ e a linguagem C# que segundo Deitel e Deitel (2016) é uma linguagem orientada a objetos que foi primeiramente anunciada pela Microsoft nos anos 2000 e tem suas raízes em linguagens como C, C++ e Java. Conjuntamente com a linguagem também são disponibilizados os frameworks .NET, que assim como Java, possui uma máquina virtual o Common Language Runtime (CLR), que é responsável por gerenciar a execução do código escrito em C#.

⁵ MICROSOFT. Visual Studio Community. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/community/>>. Acesso em: 15 nov. 2021.

Também foi empregada a ferramenta Postman⁶, gratuita para times de até três pessoas, que emula requisições HTTP de forma que é possível customizar o corpo da mensagem, o que facilita o teste e desenvolvimento de aplicações de back-end.

E para a persistência de dados, foi usado o banco de dados relacional PostgreSQL⁷ em conjunto com a ferramenta gratuita de gerenciamento de banco de dados DBeaver⁸.

3.1.3 Front-end

Para o front-end foi utilizada a IDE gratuita Visual Studio Code⁹ e o framework Angular¹⁰, que conforme Freeman (2020), é uma biblioteca de código aberto de Javascript mantido pelo Google.

As aplicações escritas em Angular usam TypeScript, que por sua vez é uma linguagem de tipagem forte escrita em cima do JavaScript. Por conta da utilização dessa, as aplicações escritas em Angular têm suporte a tipos estáticos, o que faz com que o framework fique mais próximo a linguagens como C# e Java.

O framework utiliza o Node.js¹¹, que possibilita a execução de códigos Javascript fora do ambiente de navegador. Desta forma, o Node.js se torna fundamental para o desenvolvimento de aplicações web, de forma que esse fica responsabilizado por fazer o gerenciamento dos pacotes que o desenvolvedor pode utilizar quando desenvolve uma aplicação em Angular.

3.1.4 Infraestrutura

Como o sistema desenvolvido neste trabalho está compartimentado em microsserviços foi utilizado a ferramenta Docker, desenvolvida pela Docker, Inc.¹², que consiste em uma tecnologia de conteinerização que encapsula aplicações em sistemas virtuais isolados análogos a máquinas virtuais. Schenker (2020) faz a analogia de contêineres Docker com contêineres de transporte de itens físicos, como navios de carga. O autor explica que assim como os contêineres de transporte

⁶ POSTMAN. Disponível em: <<https://www.postman.com>>. Acesso em: 17 nov. 2021.

⁷ POSTGRESQL. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 16 nov. 2021.

⁸ DBEAVER. Disponível em: <<https://dbeaver.io>>. Acesso em: 16 nov. 2021.

⁹ VISUAL STUDIO CODE. Disponível em: <<https://code.visualstudio.com>>. Acesso em: 19 nov. 2021.

¹⁰ ANGULAR. Disponível em: <<https://angular.io>>. Acesso em: 19 nov. 2021.

¹¹ NODE JS. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 20 nov. 2021.

¹² DOCKER. Disponível em: <<https://www.docker.com/company>>. Acesso em: 17 nov. 2021.

padronizaram toda a área de transporte de carga, a padronização com o Docker visa trazer a mesma revolução.

Ainda segundo Schenker, é importante salientar que apesar de análogo a virtualização através do Docker possui diferenças para com a de máquinas virtuais. A ferramenta encapsula a aplicação e as dependências externas que ela precisa para funcionar, enquanto uma máquina virtual possui todo um sistema operacional que irá ser em grande parte inutilizado no contexto de isolamento de uma aplicação. Desta forma, contêineres Docker são mais leves e flexíveis que máquinas virtuais.

O autor ainda aponta para o importante fato de que um contêiner Docker é compatível com qualquer máquina que seja compatível com o Docker Engine, indiferente do que está rodando dentro do contêiner. Desta forma, a ferramenta tem se tornado essencial para qualquer implantação de sistemas em nuvem, pois ela é quase agnóstica ao hardware.

3.1.5 Demais Ferramentas

Para o versionamento e controle do código-fonte do sistema utilizou-se o Git¹³ através da ferramenta gratuita SourceTree¹⁴, que disponibiliza uma interface gráfica intuitiva para realização da tarefa. Os repositórios do software desenvolvido neste projeto foram armazenados no GitHub¹⁵. Para a criação dos diagramas foi utilizada a ferramenta Astah UML¹⁶ que possui uma versão gratuita para estudantes. E na produção dos protótipos de tela foi utilizada a ferramenta Whimsical¹⁷, que também traz uma opção de plano gratuito.

3.2 DESENVOLVIMENTO ÁGIL DE SOFTWARE

Segundo Pressman (2011), um processo de desenvolvimento ágil de software deve ser adaptável e incremental, o que consiste em fornecer protótipos executáveis em um intervalo curto de tempo, de forma que esses se ajustem a eventuais modificações. Isso permite que o cliente avalie o sistema regularmente, ofereça seu

¹³ GIT. Disponível em: <<https://git-scm.com>>. Acesso em: 21 nov. 2021.

¹⁴ SOURCETREE. Disponível em: <<https://www.sourcetreeapp.com>>. Acesso em: 20 nov. 2021.

¹⁵ GITHUB. Disponível em: <<https://github.com>>. Acesso em: 20 nov. 2021.

¹⁶ ASTAH. Disponível em: <<https://astah.net/products/free-student-license/>>. Acesso em: 21 nov. 2021.

¹⁷ WHIMSICAL. Disponível em: <<https://whimsical.com>>. Acesso em: 21 nov. 2021.

feedback e influencie a equipe a incluir as adequações necessárias nas próximas iterações.

Essas iterações no modelo ágil Scrum são conhecidas como sprints: tarefas definidas e adaptadas aos problemas em questão, e frequentemente alteradas em tempo real. Um sprint tem como objetivo atingir um requisito em um prazo preestabelecido, e a quantidade delas em um projeto varia de acordo com o tamanho e complexidade do mesmo (PRESSMAN, 2011).

No Scrum são realizadas reuniões diárias e breves com todos os integrantes da equipe, com o foco em comunicar para os outros membros o que foi realizado, se existem obstáculos, e o que será realizado até a próxima reunião (PRESSMAN, 2011).

3.2.1 Adaptação do modelo Scrum

De início, o modelo foi adaptado conforme a necessidade da equipe e do projeto. Devido ao curto tempo para desenvolvimento do trabalho causado pela pandemia de COVID-19, o grupo optou por sprints curtas com duração de uma semana.

A equipe estabeleceu que as reuniões para definição de sprints futuras se dariam após o fim de cada iteração no aplicativo Microsoft Teams¹⁸, e as diárias ocorreriam no aplicativo de mensagens Telegram¹⁹, que também serviria para gerenciar as tarefas.

Por fim, o papel de cada um na elaboração do projeto foi estipulado. O acadêmico Raphael Pereira Rodrigues ficaria responsável pelo desenvolvimento back-end, e a acadêmica Giulia Guarise Gutierrez pela programação front-end, e ambos dividiriam as tarefas de documentação e testes no sistema.

3.2.2 Desenvolvimento do Projeto Marvin

Para a execução do sistema Marvin optou-se por segmentar seu desenvolvimento em duas etapas cronológicas: modelagem e construção.

As metas dos sprints da primeira etapa executada, a de modelagem, são apresentadas a seguir na Tabela 2, e envolvem o início do projeto, bem como seus

¹⁸ MICROSOFT. Microsoft Teams. Disponível em: <<https://www.microsoft.com/pt-br/microsoft-teams/group-chat-software/>>. Acesso em: 21 nov. 2021.

¹⁹ TELEGRAM. Disponível em: <<https://telegram.org/>>. Acesso em: 21 nov. 2021.

aspectos mais documentais, como levantamento de requisitos e realização de diagramas. O objetivo dessa etapa é modelar o projeto com base nas necessidades do cliente, e com isso auxiliar a construção do software na fase seguinte. Os detalhes de cada sprint se encontram no Apêndice 10.

TABELA 2 – METAS DOS SPRINTS DA ETAPA DE MODELAGEM

Número	Data de Início	Metas
1	27/09/2021	- Encontro com cliente do sistema - Definição do escopo do trabalho
2	04/10/2021	- Levantamento de requisitos - Diagrama de Classes
3	11/10/2021	- Validação de requisitos com o cliente - Protótipos das telas do sistema
4	18/10/2021	- Validação do cliente para Diagrama de Classes e protótipo das telas - Comparação de sistemas semelhantes
5	25/10/2021	- Diagrama de Casos de Uso - Correções no Diagrama de Classes - Correções no protótipo das telas
6	08/11/2021	- Especificação de Casos de Uso - Capítulo de Fundamentação Teórica - Protótipo: início da parte back-end
7	15/11/2021	- Diagramas de Sequência - Capítulo de Materiais e Métodos - Protótipo: finalização parte back-end; início parte front-end
8	22/11/2021	- Protótipo: finalização parte front-end; integração das partes; testes iniciais - Validação do protótipo com o cliente - Capítulo de Introdução
9	29/11/2021	- Ajustes no protótipo e testes finais - Capítulo de Apresentação do Sistema - Capítulo de Considerações Finais
10	06/12/2021	- Correções do trabalho escrito

FONTE: Os autores (2021)

As metas da etapa subsequente são mostradas na Tabela 3, e consistem majoritariamente da programação do sistema proposto. Suas intenções são a entrega de uma monografia e um software funcional que cumpra com a modelagem produzida

anteriormente, mas priorizando ainda possíveis alterações requeridas pelo dono da empresa Automar. Os detalhes de cada sprint se encontram no Apêndice 11.

TABELA 3 – METAS DOS SPRINTS DA ETAPA DE CONSTRUÇÃO

Número	Metas
11	<ul style="list-style-type: none"> - Refatoração de classes - Domínio de Cliente - Interface de Cliente - Protótipo Funcional de Orçamento
12	<ul style="list-style-type: none"> - Domínio de Cliente - Interface de Cliente - Integração do WebService ViaCEP²⁰ - Finalização e validação do Protótipo de Orçamento
13	<ul style="list-style-type: none"> - Domínio do Cliente - Integração da API Externa da Tabela FIPE²¹
14	<ul style="list-style-type: none"> - Domínio de Atendimento - Interface de Serviços
15	<ul style="list-style-type: none"> - Domínio de Atendimento - Interface de Orçamento - Validação do sistema com o cliente
16	<ul style="list-style-type: none"> - Domínio do Atendimento - Interface de Orçamento - Modificações necessárias no sistema
17	<ul style="list-style-type: none"> - Interface de Orçamento - Interface de Cliente - Interface de Agendamento - Validação do sistema com o cliente
18	<ul style="list-style-type: none"> - Interface de Agendamento - Interface de Seleção de Orçamento - Modificações necessárias no sistema
19	<ul style="list-style-type: none"> - Interface de Agendamento - Interface de Visualizar Agendamento
20	<ul style="list-style-type: none"> - Interface de Agendamento - Interface de Visualizar Agendamento - Validação do sistema com o cliente

²⁰ VIACEP. Disponível em: <<https://viacep.com.br/>>. Acesso em: 13 dez. 2021.

²¹ FORTUNA, Deivid. **Fipe API (2.0.0)**. Disponível em: <<http://deividfortuna.github.io/fipe/v2/>>. Acesso em: 13 dez. 2021.

	- Cliente requere adição de campo em telas
21	<ul style="list-style-type: none"> - Alterações nos Domínios e Interfaces para comportar novo campo - Correções e revisões na monografia
22	<ul style="list-style-type: none"> - Domínio de Autenticação - Correções e revisões na monografia
23	<ul style="list-style-type: none"> - Modificação no front-end - Correções e revisões na monografia - Validação final do sistema com o cliente
24	<ul style="list-style-type: none"> - Pequenas alterações nas interfaces - Integração do sistema - Finalização da Monografia

FONTE: Os autores (2022)

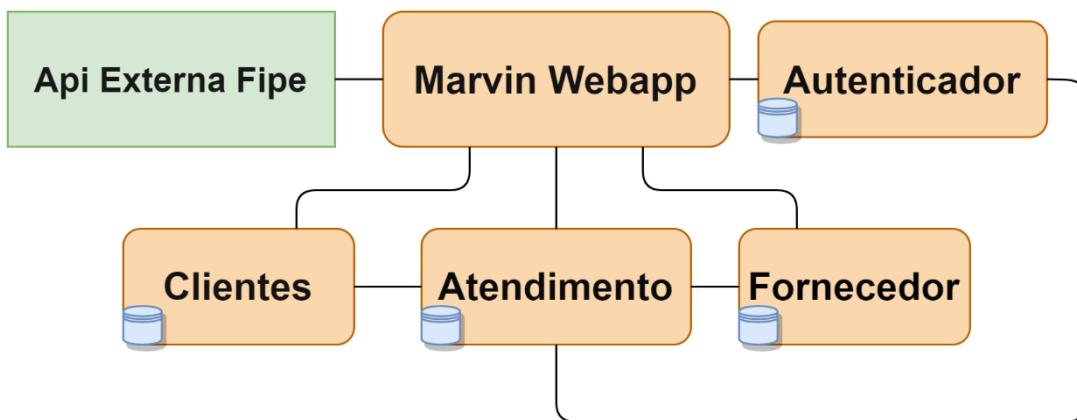
4 APRESENTAÇÃO DO SISTEMA

No presente capítulo é apresentado a arquitetura do sistema Marvin – o objeto de desenvolvimento deste trabalho –, assim como também suas dependências, apresentação das telas e funcionamento. Um breve guia de instalação do sistema está disponível no Apêndice 12.

4.1 ARQUITETURA

O sistema Marvin utiliza o padrão de projeto DDD (*Domain-Driven Design*) de forma que o proprietário da empresa Automar e os desenvolvedores possam se comunicar mais facilmente. O padrão define que seja instituído uma linguagem universal com o propósito de facilitar a determinação de domínios do sistema, bem como ser utilizada como código fonte, de forma que fique completamente explícito do que cada parte do código trata e facilite a compreensão do domínio de negócio da qual cada microserviço é responsável.

FIGURA 5 – DOMÍNIOS DO SISTEMA MARVIN



FONTE: Os autores (2022)

Desta forma os microsserviços são definidos por domínio, de forma que estes possuam responsabilidades relativamente pequenas ligadas aos domínios do negócio.

Tendo em vista essa separação de domínio, definiu-se em conjunto com o cliente diferentes domínios de negócio do sistema Marvin, como demonstrado na Figura 5.

O Marvin Webapp é a interface de usuário e a parte do sistema que agrupa e se comunica com todos os microsserviços, incluindo a API externa da Fipe²², que é completamente gratuita e serve para facilitar o cadastro de veículos e seus respectivos modelos e marcas.

A API de Clientes é responsável pelo cadastro e manutenção dos clientes e seus respectivos veículos. O domínio de Fornecedor trata do cadastro e manutenção de fornecedores e seus respectivos produtos. O microserviço de Atendimento é a API central do software que trata sobre o cerne de negócio do sistema, e o cadastro e manutenção de orçamentos e atendimentos, sendo essa API a única que se comunica com os outros serviços de forma a garantir a consistência de dados.

A API de Autenticação não tem relação direta com as regras de negócios, mas é essencial para a proteção dos dados do sistema, tendo em vista que todas as APIs do Marvin possuem autenticação via token JWT²³. Como o software não possui um

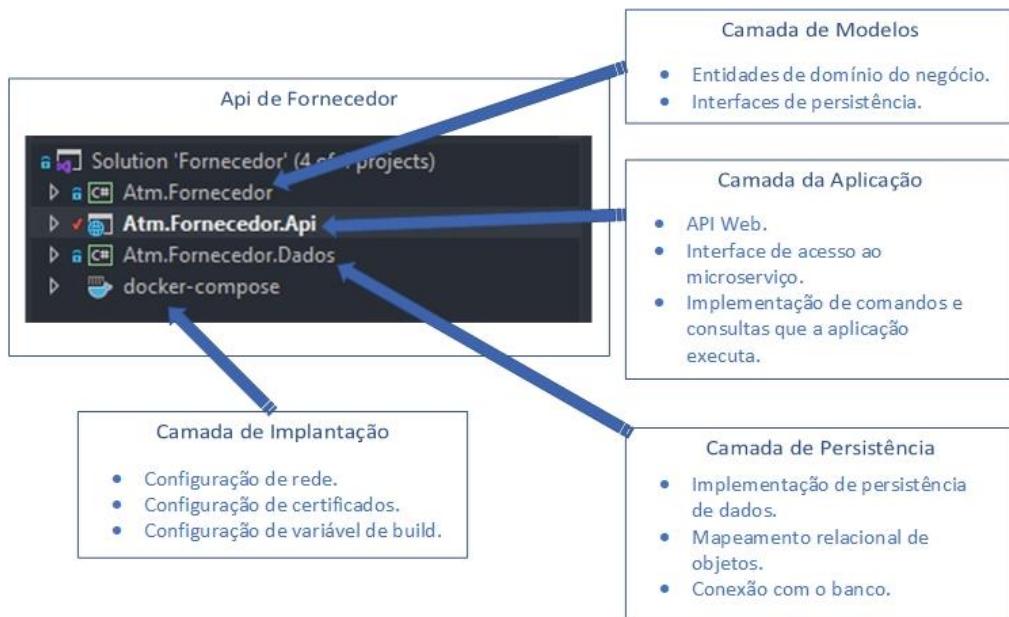
²² <https://deividfortuna.github.io/fipe/v2/>

²³ <https://datatracker.ietf.org/doc/html/rfc7519>

sistema de login, um requisito do próprio cliente, a autenticação ocorre apenas nos microsserviços e serve apenas para este propósito.

As APIs do sistema Marvin possuem o nível dois do modelo de maturidade de Richardson, que de acordo com Fowler (2010) se trata de um modelo desenvolvido por Leonard Richardson que sugere diferentes níveis de aplicações REST, e associa esses níveis ao quanto próximo uma aplicação é da definição e consegue aplicar os princípios que foram propostos por Roy Fielding na definição do padrão REST. Desta forma, as APIs desenvolvidas neste trabalho utilizam rotas separadas e bem definidas para seus respectivos recursos, assim como empregam corretamente os verbos HTTP²⁴ e seus respectivos códigos²⁵ de respostas.

FIGURA 6 – CAMADAS DO MICROSERVIÇO DE FORNECEDORES



FONTE: Os autores (2021)

No contexto da aplicação, o design orientado a domínio é utilizado para definir as responsabilidades de cada camada da aplicação. Esta separação de domínio interno da aplicação ajuda no desacoplamento e facilita a reutilização de código em outros microsserviços e agiliza o processo de desenvolvimento de software.

Nessa abordagem, a camada de modelos é responsável pela representação do negócio e deve ser desacoplada da camada de persistência, de forma que a

²⁴ <https://datatracker.ietf.org/doc/html/rfc7231#section-4.3>

²⁵ <https://datatracker.ietf.org/doc/html/rfc2616#section-10>

camada de modelos define as interfaces de persistência, mas apenas a camada de persistência as implementa.

A camada da aplicação é a responsável por toda a parte lógica das implementações, porém não pode ter responsabilidade sobre o armazenamento de informações, ela tem apenas conhecimento dos domínios e dos objetos encarregados pela persistência, faz a coordenação entre essas camadas e executa as ações.

É importante apontar que as regras de negócios foram deixadas na camada de aplicação, diferentemente do DDD tradicional onde elas são bem definidas em suas respectivas entidades. Essa abordagem diferenciada quando utilizada em sistemas distribuídos permite um maior reaproveitamento dos microsserviços, pois as regras de negócios podem ser definidas nas rotas.

A camada de persistência é responsável por fazer a conexão com o banco de dados e implementar as interfaces de persistência e mapear as entidades da camada de modelos. Já a camada de implantação se responsabiliza apenas pela configuração de implantação do próprio sistema, de modo que a configuração de infraestrutura fica mais estruturada e simplificada em forma de código.

Desta forma temos um modelo geral de aplicação que é desacoplado e altamente escalável, de modo que as diferentes camadas podem ser reutilizadas e modificadas sem muito impacto na aplicação como um todo.

Para a implementação do padrão Mediator foi utilizado a biblioteca de código aberto MediatR²⁶ (distribuído sob a licença Apache License 2.0) que facilitou imensamente a utilização do padrão Mediator e por consequência a simplificação de código e sua reutilização por todo o projeto.

Foi utilizado a versão 9.0 do C# em conjunto com o framework .NET 5.0 e também foram utilizadas as bibliotecas FluentValidation²⁷ (distribuído sob a licença Apache License 2.0) que possibilita a criação de regras de validação de tipagem forte e o Entity Framework²⁸ (distribuído sob a licença Apache License 2.0) da própria Microsoft, que foi utilizado para o mapeamento objeto-relacional, a biblioteca elimina a necessidade de escrita de SQL (Structured Query Language) e possibilita a criação de tabelas de forma automática com base nas classes de objetos do software.

²⁶ GITHUB. MediaR. Disponível em: <<https://github.com/jbogard/MediatR>>. Acesso em: 3 dez. 2021.

²⁷ FLUENT VALIDATION. Disponível em: <<https://fluentvalidation.net>>. Acesso em: 1 dez. 2021.

²⁸ MICROSOFT. Entity Framework. 2020. Não paginado. Disponível em: <<https://docs.microsoft.com/en-us/aspnet/entity-framework>>. Acesso em: 1 dez. 2021.

No sistema Marvin tanto os microsserviços, o front-end e o banco de dados estão encapsulados em contêineres, o que torna possível gerar uma imagem deste contêiner com todo o software e suas dependências. Estas imagens foram armazenadas no repositório de imagens do próprio Docker, o Docker Hub, de forma que a implantação da aplicação ficou extremamente facilitada, assim como posteriores atualizações do sistema na máquina do usuário.

Para a parte de infraestrutura foi utilizado também o Docker e através deste foi criado uma rede interna pela qual os microsserviços e o banco de dados se comunicam.

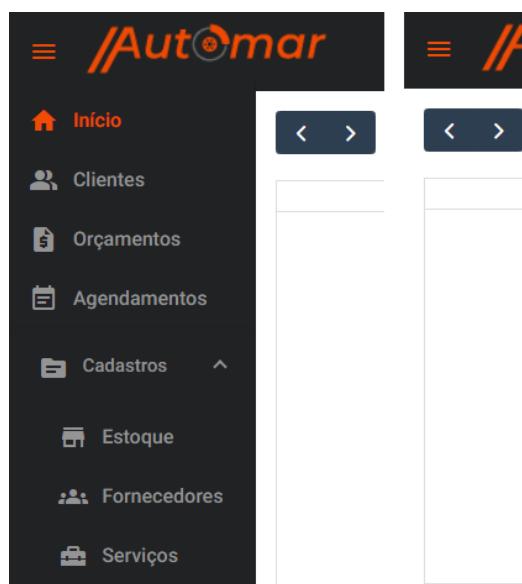
4.2 FUNCIONAMENTO

Essa seção explica as funcionalidades básicas do sistema e elucida sobre os estados internos dos componentes.

4.2.1 Menu

O menu do sistema, localizado na lateral esquerda, permite o acesso as telas principais, e pode ser recolhido e expandido utilizando o ícone de barras horizontais, posicionado no canto superior esquerdo, ao lado da logo da empresa. Ele é apresentado na Figura 7.

FIGURA 7 – MENU EXPANDIDO E RECOLHIDO



FONTE: Os autores (2022)

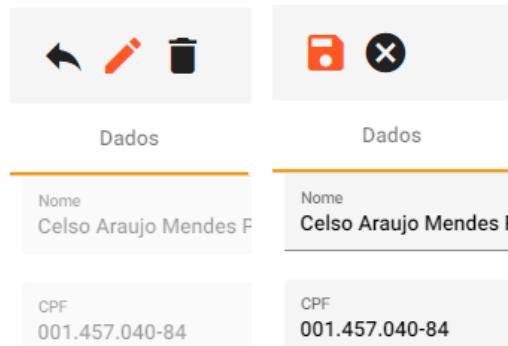
O submenu nomeado “Cadastros” também pode ser recolhido ou expandido ao clicar em cima, separando as telas de cadastro das demais.

4.2.2 Telas de Cliente, Orçamento e Agendamento

Essas telas possuem o mesmo funcionamento básico. Após a abertura, uma tabela com todos os registros armazenados é mostrada, bem como um botão (“Novo”) para criação de novo cadastro. Ao clicar duas vezes em cima de um registro (linha) o usuário é redirecionado para tela que permite a edição, exclusão e visualização completa do mesmo.

Essa interface apresenta ícones que realizam ações. O de voltar (flecha) retorna a tela de listagem; o de excluir (lixo) apaga o cadastro; e o de editar (lápis) habilita os campos para modificações e faz com que os ícones de salvar (disquete) e descartar (x) sejam exibidos, como mostra a Figura 8.

FIGURA 8 – ÍCONES EM COMUM DAS TELAS DE CLIENTE, ORÇAMENTO E AGENDAMENTO

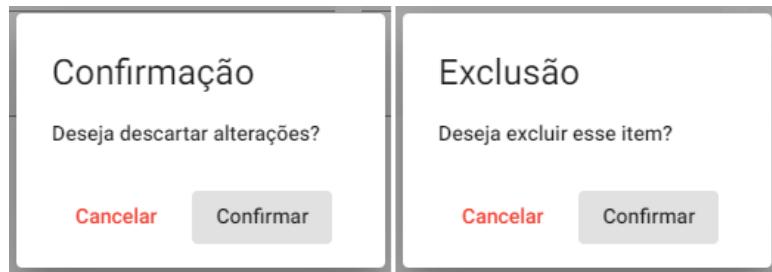


FONTE: Os autores (2022)

O ícone de salvar, como o nome já indica, grava um novo registro ou salva as alterações realizadas. O de descartar serve para cancelar um novo cadastro ou desfazer os campos alterados pelo usuário. Após a ação bem sucedida de ambos o formulário volta para o estado de visualização.

Para os ícones de descarte e exclusão uma janela de confirmação se abre, como mostra a Figura 9.

FIGURA 9 – JANELAS DE CONFIRMAÇÃO



FONTE: Os autores (2022)

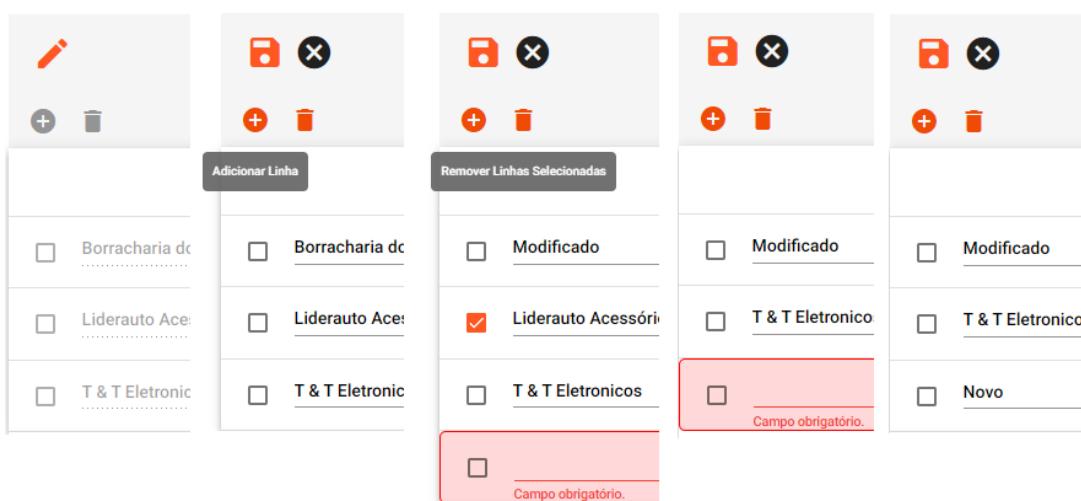
4.2.3 Telas de Cadastros

São as telas que ficam no submenu “Cadastros” dentro do menu. Seu comportamento é similar ao das telas explicadas anteriormente nos quesitos editar, salvar e desfazer, mas seu comportamento muda na criação e exclusão.

Nessas interfaces é possível cadastrar, editar e deletar mais de um registro ao mesmo tempo, necessitando que o formulário esteja habilitado (clicar no botão de editar).

O botão de adicionar linha (mais) coloca novos registros na tabela, que podem então ser modificados da mesma forma que os previamente cadastrados. Para remover uma ou mais linhas basta marcar a caixa de seleção – a coluna mais a esquerda, em formato de caixa – e clicar no botão de remover linhas selecionadas (lixeira). A Figura 10 retrata esse fluxo.

FIGURA 10 – FLUXO DE ACÕES EM TABELA

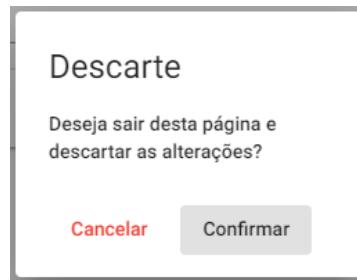


FONTE: Os autores (2022)

4.2.4 Confirmação de Mudança de Tela

Caso o usuário esteja editando um cadastro e decida ir para outra tela pelo menu uma confirmação de descarte também aparecerá, como apresenta a Figura 11.

FIGURA 11 – CONFIRMAÇÃO PARA TROCAR DE TELA

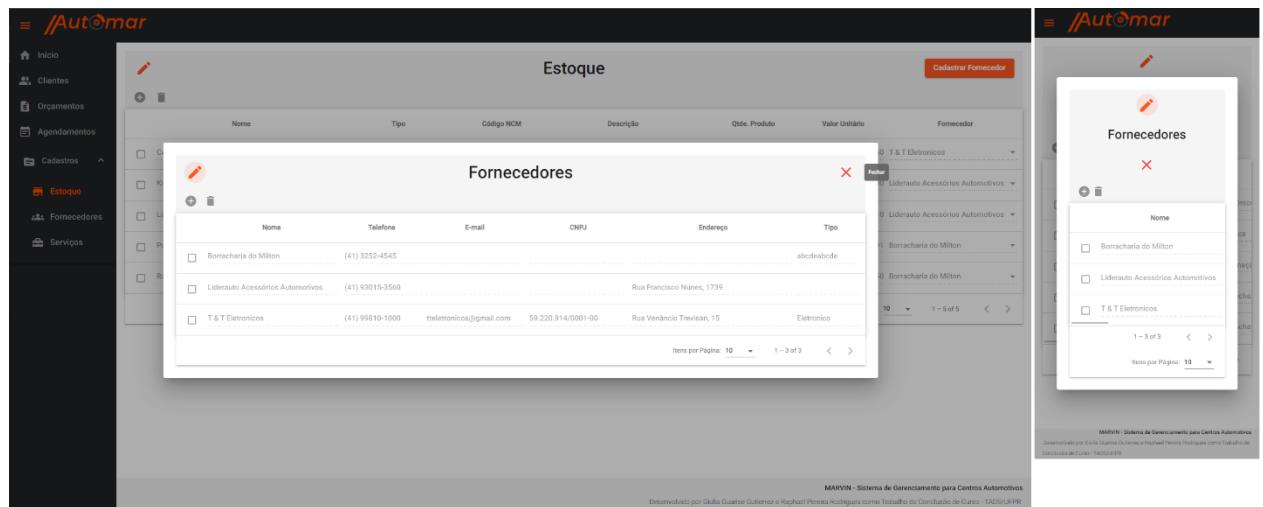


FONTE: Os autores (2022)

4.2.5 Fechar Caixas de Diálogo (Tela Suspensa)

As caixas de diálogo podem ser em formato de confirmação, como visto anteriormente, ou apresentar uma tela suspensa. No último caso, para fechá-las basta clicar no botão “X” em vermelho, localizado no canto superior direito. Um exemplo pode ser visto na Figura 12.

FIGURA 12 – EXEMPLO DE JANELA DE CADASTRO SUSPENSA



FONTE: Os autores (2022)

4.2.6 Estados Internos dos Componentes

A Tabela 4 abaixo apresenta detalhes sobre os estados de alguns componentes do sistema, que auxiliam a entender qual fluxo interno o cadastro especificado realiza.

TABELA 4 – ESTADOS INTERNOS DOS COMPONENTES

Componente	Estados	Sobre
Orçamento	1 - Cadastrado	Um orçamento se inicia com estado 1, ao agendar passa para 2, e ao finalizar 3. O usuário poderá transitar entre seus estados, sempre cumprindo a ordem numérica crescente ou decrescente.
	2 - Agendado	
	3 - Finalizado	Um orçamento com estado 2 será referido nesse documento como um agendamento.
Cliente	1 - Ativo	O usuário pode deletar um cliente no sistema, porém para garantir que um orçamento mantenha suas informações ele será apenas desativado. O cliente excluído não poderá ser restaurado.
	2 - Inativo	
Carro	1 - Ativo	Um carro pode estar associado a mais de um cliente. Se um carro não possui nenhum cliente vinculado ele passa para o estado 2.
	2 - Inativo	Ele também é desativado para manter os dados no orçamento.
Estoque	1 - Ativo	Possui os estados pelo mesmo motivo de cliente e carro, e também por conta de uma tabela do banco de dados que armazena os históricos de preço de um determinado produto (não utilizada no sistema atualmente).
	2 - Inativo	
Demais	-	Os demais componentes do sistema não possuem estado. Se uma exclusão é realizada serão removidos totalmente do banco de dados.

FONTE: Os autores (2022)

4.3 TELAS DO SISTEMA

Nesta seção, as telas do software são apresentadas simultaneamente com as funcionalidades próprias de cada uma, visto que as compartilhadas já foram apresentadas na seção passada.

O sistema, como citado nos capítulos anteriores, foi desenvolvido para o proprietário do centro automotivo Automar com o objetivo de gerenciar o atendimento ao cliente de maneira simples, objetiva e flexível.

Correspondendo a isso, as interfaces se adaptam a diferentes tamanhos de telas; otimizam a criação, visualização, edição e exclusão de cadastros básicos; concentram operações dependentes em uma única tela, para que o usuário não precise cancelar uma ação por esquecer de registrar uma informação previamente; e possuem poucos campos com obrigatoriedade.

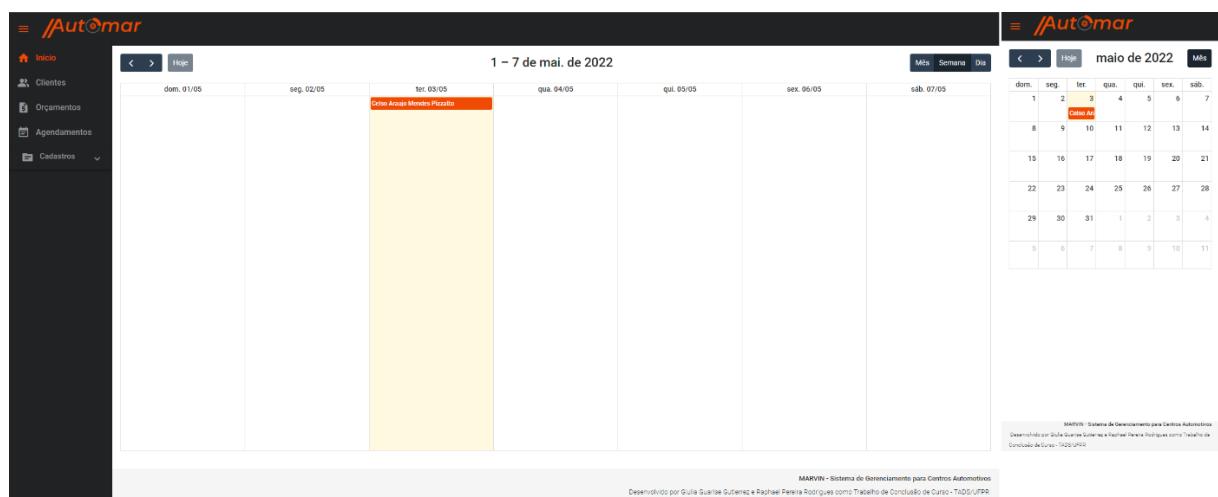
Vale ressaltar que o cliente especificou que não houvesse um login para a aplicação, visto que ele entende como um empecilho para exercer suas atividades.

Porém, como descrito na seção de Arquitetura, existe uma autenticação para validar as transações, oculta ao usuário.

4.3.1 Tela de Início

A tela inicial do sistema contém um calendário com os agendamentos cadastrados. Em dispositivos móveis apenas a configuração para calendário mensal aparece, enquanto para os demais o usuário poderá escolher entre exibição mensal, semanal ou diária, como mostra a Figura 13.

FIGURA 13 – TELA DE INÍCIO



FONTE: Os autores (2022)

Ao clicar em um determinado dia do calendário que possua eventos é aberta uma janela com uma tabela de detalhes – que também permite a finalização de um agendamento listado – e uma opção de impressão, como apresenta a Figura 14. Ao clicar duas vezes em uma linha, o usuário será redirecionado para tela do agendamento escolhido para ter acesso à todas as informações do mesmo.

FIGURA 14 – TABELA DE AGENDAMENTOS DIÁRIOS

The screenshot shows the MARVIN system interface. At the top, there's a navigation bar with links for 'Início', 'Clientes', 'Orçamentos', 'Agendamentos', and 'Cadastros'. Below this is a date range selector '1 - 7 de mai. de 2022' and a 'Mês' dropdown. The main area displays a weekly calendar grid from Sunday to Saturday. A specific date, 'ter 03/05', is highlighted in red and contains the text 'Celso Araújo Mendes Pizzatto'. A modal window titled 'Agendamentos' is open over the calendar, showing a table with columns: Status, Cliente, Carro, Valor Final, and Hora Marcada. One row is populated with 'Agendado', 'Celso Araújo Mendes Pizzatto', 'AAE-1C52', 'R\$ 608,81', and '09:40'. At the bottom of the modal is a button labeled 'Criar Agendamento'. In the bottom right corner of the main screen, there's a small preview of another window titled 'Agendamentos' showing a list of completed appointments.

FONTE: Os autores (2022)

Ao clicar em uma data sem eventos a interface sugere a criação de um novo agendamento.

A impressão supracitada será utilizada pelo usuário no final do expediente como forma de resumo dos atendimentos completados com sucesso no dia, ou seja, apenas os finalizados. A Figura 15 apresenta a parte preenchida do relatório gerado.

FIGURA 15 – RELATÓRIO DIÁRIO DE AGENDAMENTOS FINALIZADOS

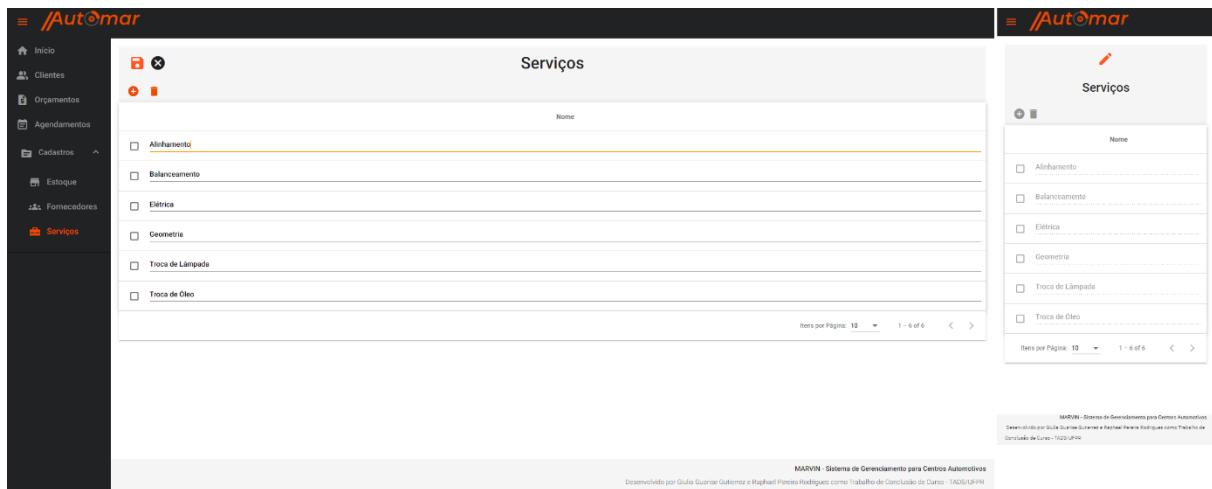
Relatório de Agendamentos Finalizados 03/05/2022			
Quantidade de Agendamentos: 1		Emitido em: 04/05/2022 19:12:00	
Cliente	Carro	Valor Final	Hora Marcada
Celso Araújo Mendes Pizzatto	AAE-1C52	R\$ 608,81	09:40

FONTE: Os autores (2022)

4.3.2 Tela de Cadastro de Serviços

Nesse contexto, um serviço é apenas um texto que serve para facilitar e padronizar o cadastro de orçamento, como apresenta a Figura 16.

FIGURA 16 – TELA DE CADASTRO DE SERVIÇOS

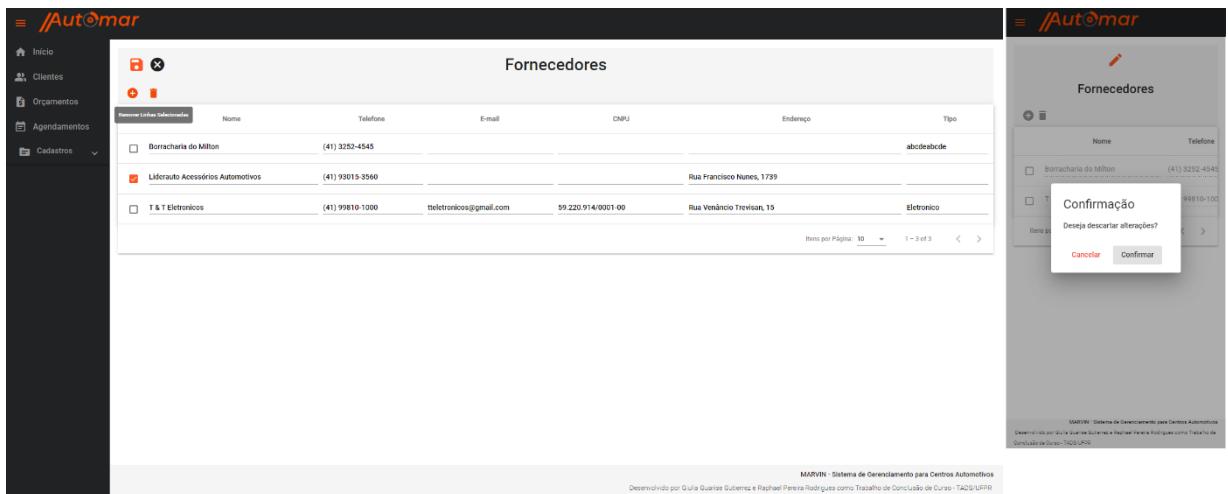


FONTE: Os autores (2022)

4.3.3 Tela de Cadastro de Fornecedores

A Figura 17 mostra que nesta tela o usuário pode armazenar informações como nome, telefone, e-mail e endereço de fornecedores, para fins de consultas futuras e associação com um item em estoque.

FIGURA 17 – TELA DE CADASTRO DE FORNECEDORES



FONTE: Os autores (2022)

4.3.4 Tela de Cadastro de Estoque

O estoque será composto por qualquer item que seja previamente adquirido pelo proprietário da Automar. Um fornecedor deve ser obrigatoriamente associado a um produto.

Nessa tela (Figura 18) também é possível cadastrar um fornecedor sem perder as modificações já realizadas por meio da janela suspensa, como mostra a Figura 12 da seção anterior.

FIGURA 18 – TELA DE CADASTRO DE ESTOQUE

Nome	Tipo	Código NCM	Descrição	Qtd. Produto	Valor Unitário	Fornecedor
Central Multimídia MPS	Eletrônico	36511202	Touch Screen com bluetooth	1	R\$ 373,50	T & T Eletrônicos
Kit Travas Elétricas	Elétrica	12441247	Tech One Universal Duplo Comando para 4 Portas	3	R\$ 59,90	Liderauto Acessórios Automotivos
Lâmpada H4 Super Led	Iluminação	97745644		4	R\$ 25,10	Liderauto Acessórios Automotivos
Pneu Firestone	Borracharia	04547841	F-Series F-600 P 195/55R15 85 H	2	R\$ 350,91	Borracharia do Milton
Rosca para bico de pneu	Borracharia	12423943		15	R\$ 9,90	Borracharia do Milton

FONTE: Os autores (2022)

4.3.5 Telas de Listagem

Essas telas possuem uma tabela com a lista de dados cadastrados, com apenas algumas informações básicas para diferenciar cada linha que está sendo mostrada. Também contêm o botão para criar um novo registro.

As Figuras 19, 20 e 21 apresentam, respectivamente, as telas de listagens de cliente, orçamento e agendamento.

FIGURA 19 – TELA DE LISTAGEM DE CLIENTES

Nome	CPF	Telefone
Celso Araújo Mendes Pizzatto	001.457.040-84	(41) 3454-2455
Maria da Silva	047.286.749-08	(41) 99999-9999
Terezinha Oliveira	884.771.000-63	

FONTE: Os autores (2022)

FIGURA 20 – TELA DE LISTAGEM DE ORÇAMENTOS

FONTE: Os autores (2022)

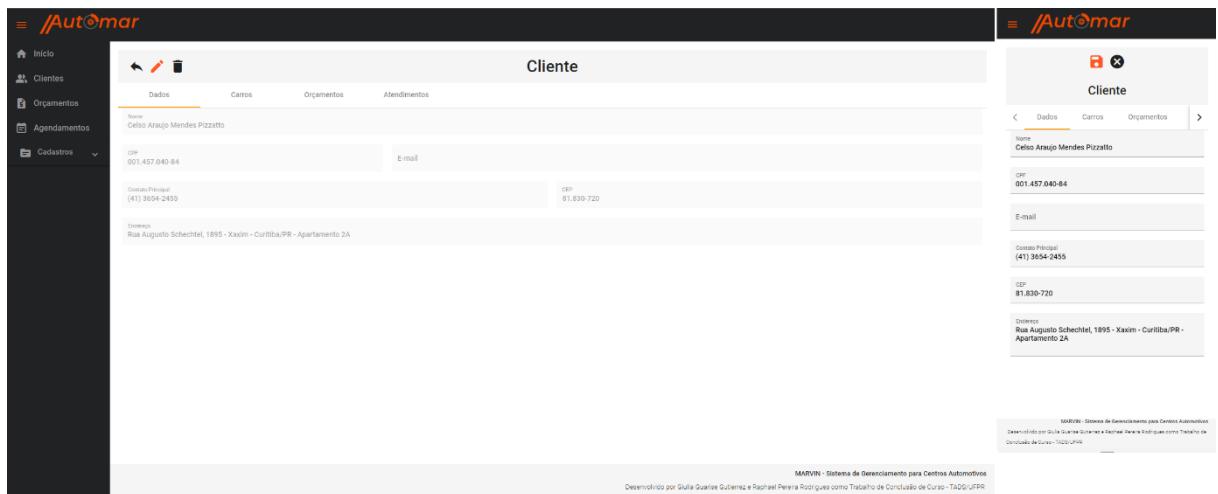
FIGURA 21 – TELA DE LISTAGEM DE AGENDAMENTOS

FONTE: Os autores (2022)

4.3.6 Tela de Cliente

A primeira aba dessa tela (“Dados”) possui os dados cadastrais do cliente, como mostra a Figura 22.

FIGURA 22 – TELA DE CLIENTE – ABA DE DADOS

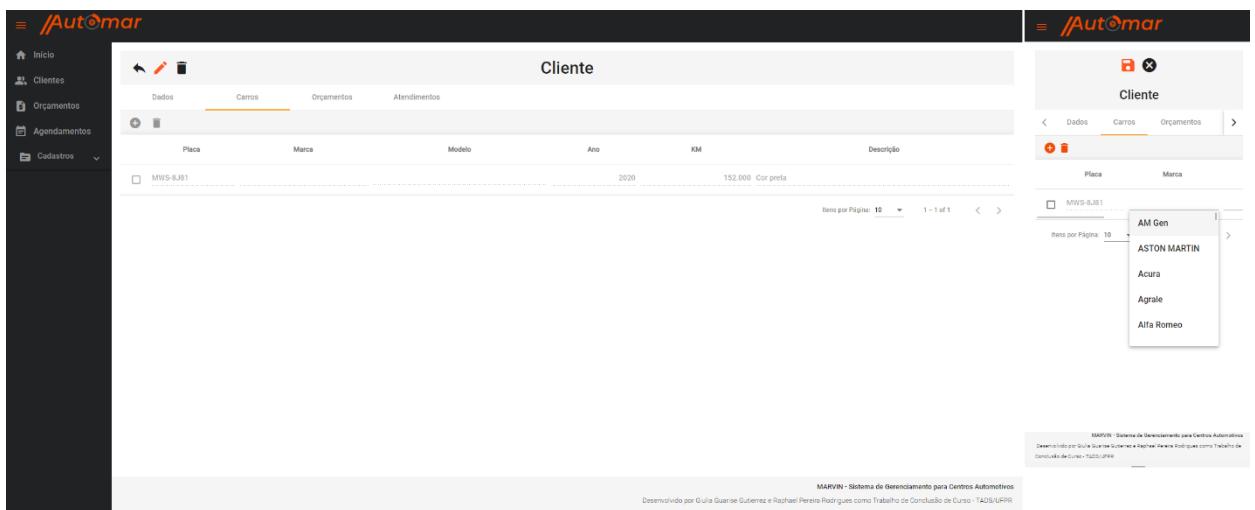


FONTE: Os autores (2022)

Na aba de Carros, os veículos associados àquele cliente são adicionados, como apresenta a Figura 23. O funcionamento da tabela é igual às das telas de cadastro (por exemplo, a tela de cadastro de serviços).

Um cliente pode ter vários carros, e um carro pode pertencer a vários clientes – nesse caso, alterar o cadastro do automóvel em um cliente refletirá para o outro.

FIGURA 23 – TELA DE CLIENTE – ABA DE CARROS



FONTE: Os autores (2022)

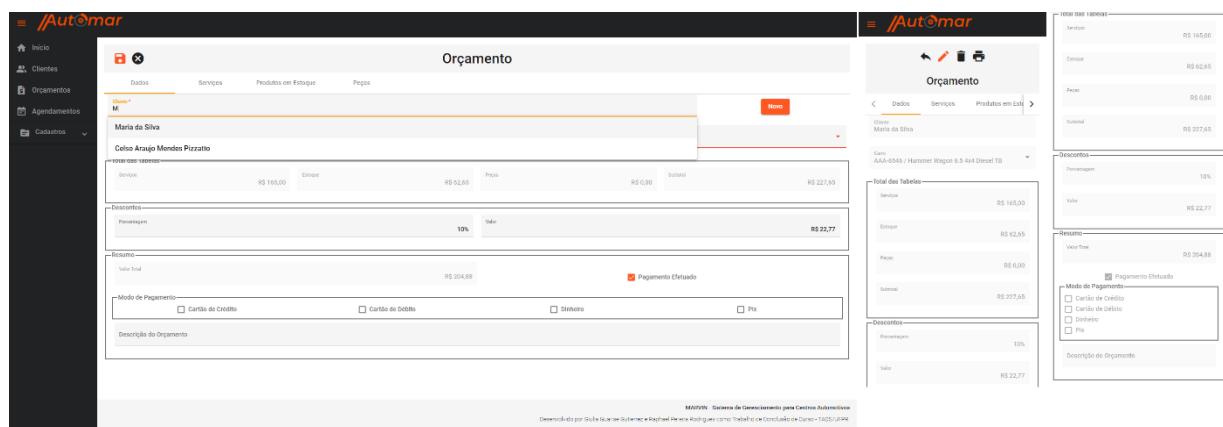
A tela de cliente também possui a aba de “Orçamento”, com os orçamentos realizados para o cliente, e “Agendamentos”, com os agendamentos em aberto ou finalizados do mesmo.

4.3.7 Tela de Orçamento

Na primeira aba, apresentada na Figura 24, é escolhido um cliente com seu respectivo carro, sendo possível cadastrar um novo cliente com seus automóveis em tela. Na divisória intitulada “Total das Tabelas” é mostrada a somatória dos honorários dos serviços e dos valores dos itens em estoque e peças cadastrados nas outras abas.

Na divisória “Descontos” o usuário pode optar por dar um desconto em porcentagem ou valor, que será subtraído do subtotal calculado na divisória anterior. Por fim, na divisória “Resumo” o valor total é disponibilizado e detalhes sobre o pagamento e orçamento podem ser informados.

FIGURA 24 – TELA DE ORÇAMENTO – ABA DE DADOS



FONTE: Os autores (2022)

Na aba de Serviços o usuário seleciona um dos serviços previamente cadastrados – sendo viável fazê-lo em tela pelo botão “Editar Serviços” – e informa o honorário a ser cobrado, podendo adicionar uma descrição para melhor especificar o que será realizado, como mostra a Figura 25.

FIGURA 25 – TELA DE ORÇAMENTO – ABA DE SERVIÇOS

Serviço	Descrição	Honorário
Balanceamento		R\$ 45,00
Alinhamento		R\$ 45,00
Troc de Lâmpada	Lâmpadas dianteiras	R\$ 75,00

FONTE: Os autores (2022)

Na seguinte, escolhe os produtos em estoque que serão utilizados para o atendimento ao cliente. O valor unitário é preenchido de acordo com o cadastro prévio, e o usuário indica o percentual de lucro que quer ter em cima do valor e a quantidade que será utilizada, como apresenta a Figura 26.

FIGURA 26 – TELA DE ORÇAMENTO – ABA DE PRODUTOS EM ESTOQUE

Produto	Percentual de Lucro	Valor Unitário	Quantidade	Total
Lâmpada H4 Super Led	10%	R\$ 27,61	2	R\$ 55,22
Rosca para bico de pneu	35%	R\$ 7,42	1	R\$ 7,42

FONTE: Os autores (2022)

É possível cadastrar os produtos – e até mesmo fornecedores – ao clicar no botão de “Editar Estoque”, sem precisar descartar as modificações realizadas.

A Figura 27 expõe a última aba, que se referem aos itens (aqui nominados de peças) comprados conforme demanda, por serem caros ou para uma marca e modelo de carro muito específica.

FIGURA 27 – TELA DE ORÇAMENTO – ABA DE PEÇAS

Nome	Descrição	Código NCM	Valor de Compra	Percentual de Lucro	Valor de Venda	Quantidade	Valor Total
Têxile Automotivo		55132333	R\$ 50,00	5%	R\$ 52,50	1	R\$ 52,50

FONTE: Os autores (2022)

O botão de impressão emite uma ordem de serviço como a da Figura 28, que como explicado em capítulos anteriores irá oficializar o serviço a ser prestado.

FIGURA 28 – RELATÓRIO DE ORDEM DE SERVIÇO

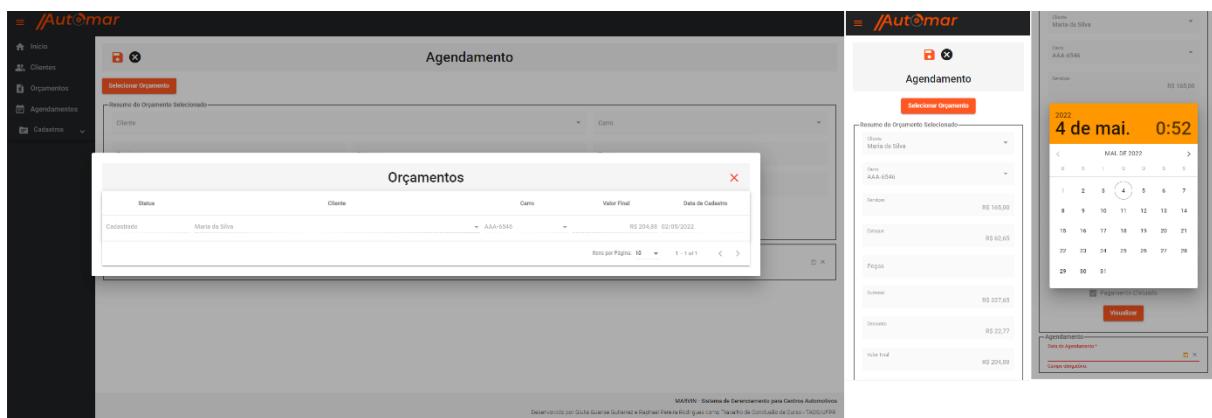
AUTOMAR CENTRO AUTOMOTIVO		CNPJ: 77.638.245/0001-73	
		Rua Anita Ribas, 39 - Curitiba, PR	
		Contato: (41) 98904-7416	
Identificador: 0a2a47d2 Emitido em: 05/05/2022 Válido até: ___/___/___			
CLIENTE / CARRO			
Nome	Maria da Silva	CPF	047.286.749-08
E-mail	maria@gmail.com	Telefone	(41) 99999-9999
Endereço	Rua Marcondes de Oliveira, 145 - Colonial - Londrina/PR		
Marca / Modelo	AM Gen Hummer Wagon 6.5 4x4 Diesel TB	Placa	AAA-6546
ORÇAMENTO			
Produto / Serviço	Valor Unitário	Quantidade	Total
Balancamento	R\$ 45,00	1	R\$ 45,00
Alinhamento	R\$ 45,00	1	R\$ 45,00
Troca de Lâmpada	R\$ 75,00	1	R\$ 75,00
Lâmpada H4 Super Led	R\$ 27,61	2	R\$ 55,22
Rosca para bico de pneu	R\$ 7,43	1	R\$ 7,43
Subtotal: R\$ 227,65		Desconto: R\$ 22,77	Total: R\$ 204,88
OBSERVAÇÕES			
Forma(s) de Pagamento: Pagamento efetuado: Sim Descrição:			
_____ Maria da Silva			

FONTE: Os autores (2022)

4.3.8 Tela de Agendamento

Na criação de um novo agendamento a tela da Figura 29 é mostrada. Ao clicar em “Selecionar um orçamento” uma janela suspensa aparece e após selecionado (com duplo clique) os dados em tela são preenchidos e a data de agendamento é habilitada.

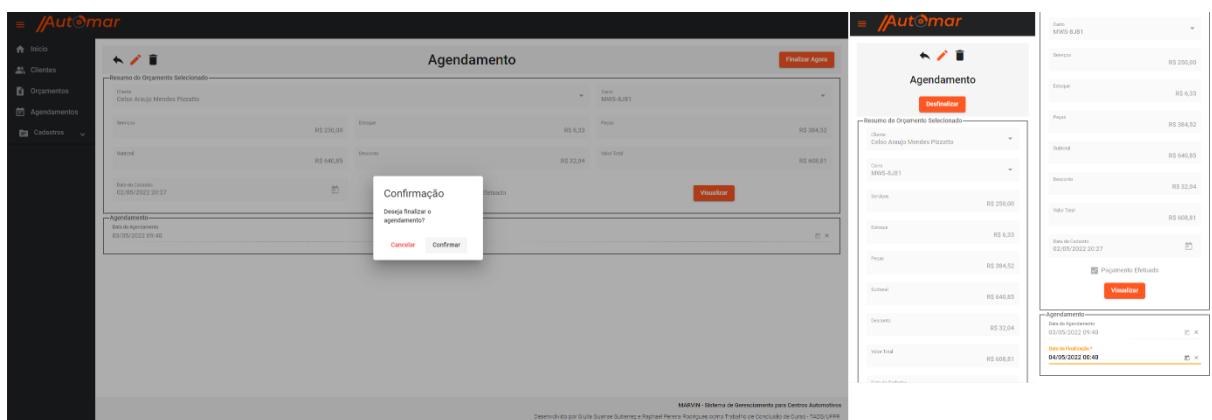
FIGURA 29 – TELA DE AGENDAMENTO - NOVO



FONTE: Os autores (2022)

Um agendamento pode ser finalizado ou desfeito (desfazer a finalização), como apresenta a Figura 30. Sua data de finalização também poderá ser editada após a ação ocorrer.

FIGURA 30 – TELA DE AGENDAMENTO – FINALIZAR E “DESFINALIZAR”



FONTE: Os autores (2022)

5 CONSIDERAÇÕES FINAIS

A finalidade principal deste projeto foi o desenvolvimento escalável e flexível de um Sistema de Gerenciamento para Centros Automotivos, visando atender as necessidades do dono da empresa Automar.

No início do desenvolvimento do projeto, foram realizadas reuniões com o cliente do Marvin – nome atribuído ao software – para entender suas demandas específicas e captar os processos envolvidos no atendimento ao cliente em um centro automotivo. Baseado nisso, a ideia de um sistema escalável e adaptável nasceu, junto com a premissa de descomplicar e otimizar as interfaces para acompanhar o ritmo diário da referida empresa.

Após, utilizou-se majoritariamente pesquisas de sistemas similares, elaboração de diagramas e feedback do cliente na construção da base para atingir o objetivo, bem como a implementação da prova de conceito (protótipo) das funcionalidades de Cadastro de Fornecedor e Cadastro de Produtos.

Todas essas fases do projeto, bem como as futuras, foram organizadas utilizando uma adaptação da metodologia ágil Scrum, com sprints definindo as metas para cada semana.

O desenvolvimento do sistema ocorreu de maneira constante, com maior ênfase em receber as avaliações do cliente assim que uma nova funcionalidade era implementada, o que evita a propagação de erros e garante a qualidade do software.

Dito isso, embasando-se no feedback do cliente e nas análises da equipe desenvolvedora, o Marvin atinge os requisitos e objetivos traçados, pois é um sistema com interfaces fluidas que pode ser escalado futuramente, podendo ser utilizado por outros centros automotivos de pequeno porte.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

As seguintes sugestões podem ser integradas ao sistema Marvin à medida em que o sistema for implantado em nuvem e conforme a demanda de crescimento do mesmo:

- Implementar sistema de login para vários usuários;

- Integrar os repositórios das soluções com alguma plataforma de integração contínua, como o Azure DevOps²⁹;
- Utilizar o Traefik³⁰ como proxy-reverso para roteamento entre microsserviços;
- Migrar as soluções do Docker para algum orquestrador de contêineres, como Kubernetes³¹;
- Separar o domínio de Pagamento do domínio de Atendimento para maior desacoplamento do sistema e assincronicidade caso o sistema seja integrado com plataforma externa de pagamentos;
- Criação de uma API Gateway para maior desacoplamento dos microsserviços e da interface de usuário, permitindo assim um maior reaproveitamento dos sistemas;
- Integração com algum software de mensagens, como RabbitMQ³², para a comunicação assíncrona entre microsserviços e melhor performance do sistema como um todo.

²⁹ AZURE DEVOPS. Disponível em: <<https://azure.microsoft.com/pt-br/services/devops/>> Acesso em: 12 dez. 2021.

³⁰ TRAEFIK. Disponível em: <<https://doc.traefik.io/traefik/>> Acesso em: 12 dez. 2021.

³¹ KUBERNETES. Disponível em: <<https://kubernetes.io/pt-br/>> Acesso em: 12 dez. 2021.

³² <https://www.rabbitmq.com>

REFERÊNCIAS

BERTRAND, M. **Object-oriented Software Construction**. Prentice-Hall, Abril 13, 1997.

DEITEL, P.; DEITEL, H. **C# 6 for Programmers**. Pearson Education, Inc. Agosto, 2016.

EVANS, E. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. Addison-Wesley Professional, Agosto 20, 2003.

FREEMAN, A. **Angular 9 Build Powerful and Dynamic Web Apps**. Apress. 2020.

GAI, S. **Building a Future-Proof Cloud Infrastructure: A Unified Architecture for Network, Security and Storage Services**. Addison-Wesley Professional, 2020.

GAMMA, E.; HELM, R.; JOHNSON, R. VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison Wesley Longman, Inc. Agosto, 1997.

MARTIN, C. R. **Agile Software Development: Principles, Patterns, and Practices**. Pearson Education, 2003.

MDN WEB DOCS. **Debugger**. 2021. Não paginado. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/debugger>>. Acesso em: 22 nov. 2021.

NADER, D. Ordem de serviço: o que é, para que serve e como emitir? **Contábeis**, 2020. Disponível em: <<https://www.contabeis.com.br/noticias/44488/ordem-de-servico-o-que-e-para-que-serve-e-como-emitir>>. Acesso em: 23 nov. 2021.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 7.ed. Porto Alegre: MacGraw-Hill, 2011. Disponível em: <https://www.academia.edu/41207478/Engenharia_de_Software_Uma_Abordagem_Profissional>. Acesso em: 5 dez. 2021.

SCHENKER, N. G. **Docker Fundamentals of Docker 19.x**. Packt Publishing, Ltd. Março, 2020.

ULTRACAR. **O que é um auto center e como abrir um? 4 dicas para começar com o pé direito**. 2019. Não paginado. Disponível em: <<https://ultracar.com.br/sistema-gestao-oficina-mecanica/programa-gerenciamento-oficina-mecanica/o-que-e-um-auto-center-e-como-abrir-um/>>. Acesso em: 22 nov. 2021.

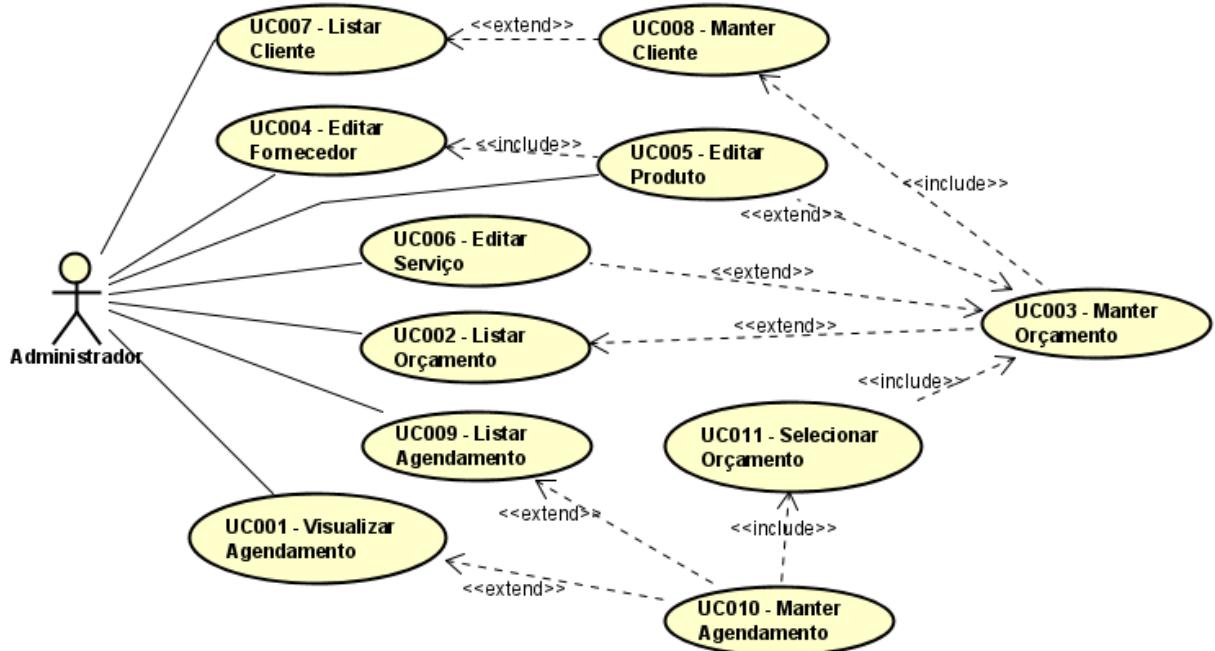
FOWLER, M. **Richardson Maturity Model: steps toward the glory of REST**. 2010. Disponível em: <<https://martinfowler.com/articles/richardsonMaturityModel.html>>. Acesso em: 3 Maio, 2022.

APÊNDICE 1 – LISTA DE REQUISITOS FUNCIONAIS

- RF1 – O sistema deve permitir cadastrar e manter clientes e seus respectivos veículos.
- RF2 – No cadastro clientes é obrigatório o vínculo com ao menos um veículo.
- RF3 – No cadastro de cliente apenas o nome deve ser obrigatório.
- RF4 – No cadastro de veículo apenas a placa deve ser obrigatória.
- RF5 – O sistema deve permitir cadastrar e manter fornecedores e seus respectivos produtos.
- RF6 – No cadastro de fornecedor apenas o nome deve ser obrigatório.
- RF7 – No cadastro de produto, esse deve ser obrigatoriamente vinculado a um fornecedor.
- RF8 – O sistema deve permitir cadastrar e manter orçamentos de clientes.
- RF9 – O sistema deve disponibilizar um modo de impressão de orçamento com campo de assinatura do cliente.
- RF10 – O sistema deve possuir um modo de agendar data de atendimento referente a um específico orçamento.
- RF11 – O sistema deve permitir cadastrar e manter serviços.
- RF12 – O sistema deve permitir cadastrar peças e valores de serviços referentes a um orçamento durante o cadastro de um dado orçamento.
- RF13 – No cadastro de orçamento é obrigatório que ao menos um serviço, produto ou peça seja cadastrado.
- RF14 – O sistema deve possuir um calendário organizado pelas respectivas datas de orçamentos.
- RF15 – O sistema deve ter a opção de impressão em lista dos agendamentos finalizados do dia.
- RF16 – O cadastro de produto ou peça deve conter seu código NCM (Nomenclatura Comum do Mercosul) para facilitar a emissão posterior de nota fiscal.
- RF17 – O sistema não deve possuir uma tela de login.

APÊNDICE 2 – DIAGRAMA DE CASOS DE USO

FIGURA 31 – DIAGRAMA DE CASOS DE USO



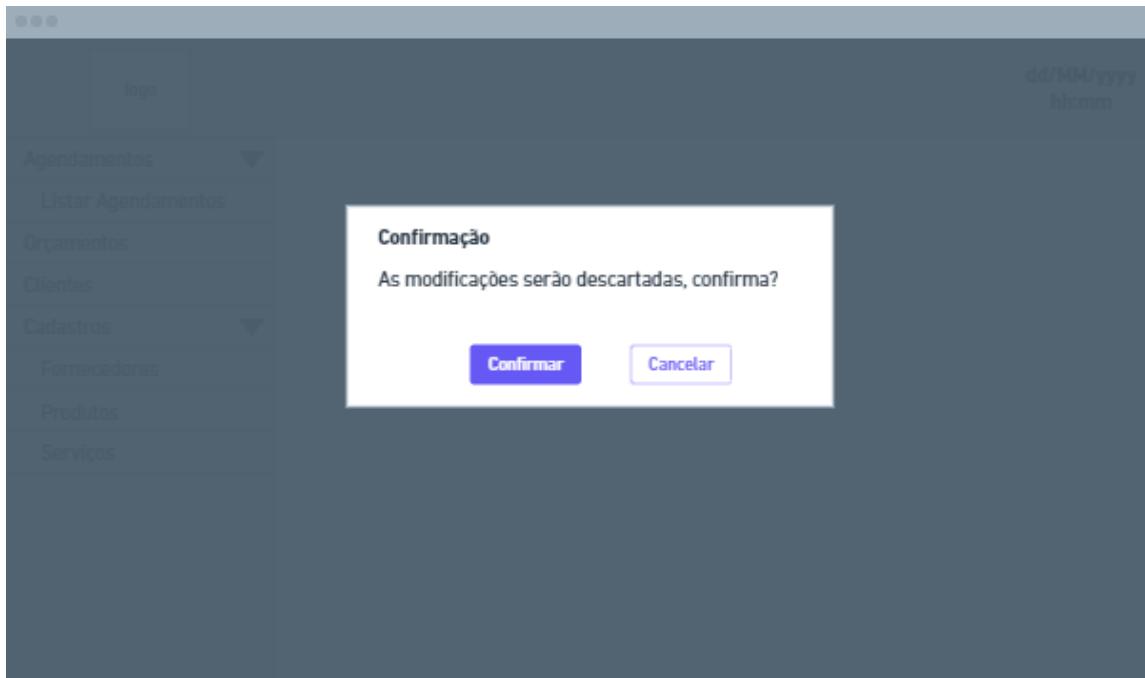
Fonte: Os autores (2021)

APÊNDICE 3 – LISTA DE ÍCONES DAS TELAS

-  - botão “Salvar” desabilitado
-  - botão “Salvar” habilitado
-  - botão “Cancelar” desabilitado
-  - botão “Cancelar” habilitado
-  - botão “Editar” desabilitado
-  - botão “Editar” habilitado
-  - botão “Excluir” desabilitado
-  - botão “Excluir” habilitado
-  - botão “Imprimir” desabilitado
-  - botão “Imprimir” habilitado

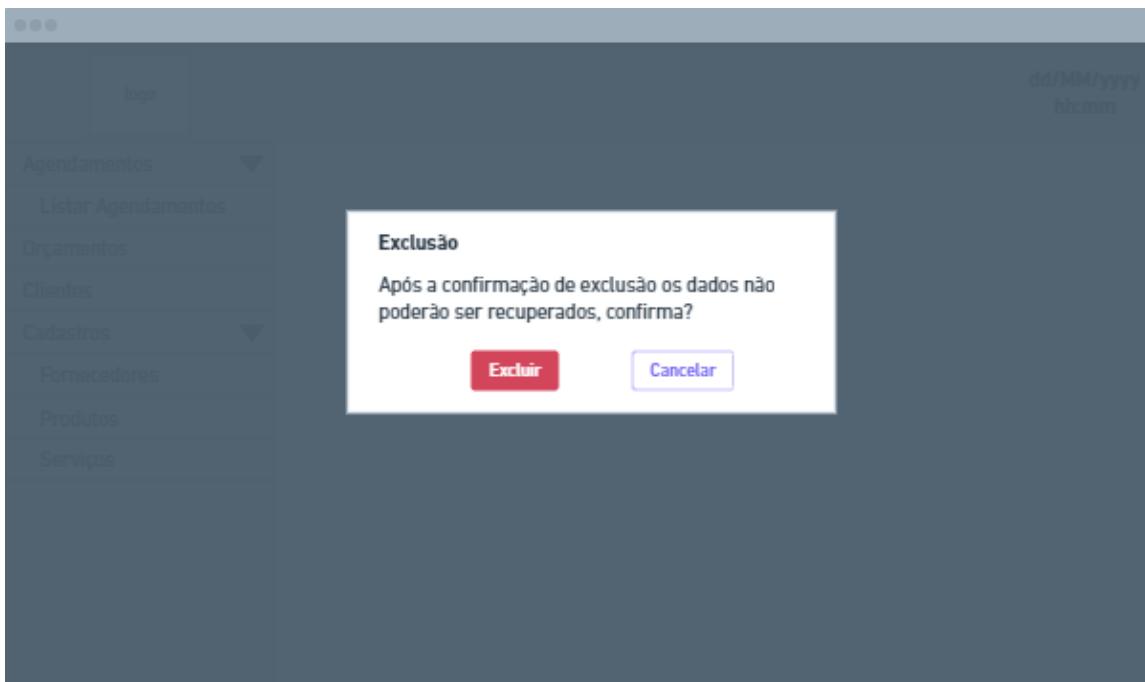
APÊNDICE 4 – TELAS GENÉRICAS DO SISTEMA

FIGURA 32 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR



FONTE: Os autores (2021)

FIGURA 33 - TELA DE CONFIRMAÇÃO DO BOTÃO EXCLUIR



FONTE: Os autores (2021)

APÊNDICE 5 – ESPECIFICAÇÃO DE CASOS DE USO

UC001 – Visualizar Agendamento

Descrição

Este caso de uso serve para mostrar a quantidade de agendamentos e seus resumos em cada dia do mês.

Data View

DV1 – Tela de agendamentos

The screenshot shows a user interface for managing appointments. On the left is a sidebar with navigation links: 'Agendamentos' (selected), 'Listar Agendamentos', 'Orçamentos', 'Clientes', 'Cadastros' (selected), 'Fornecedores', 'Produtos', and 'Serviços'. The main area has a header with a logo placeholder and date/time fields ('dd/MM/yyyy hh:mm'). Below the header is a button bar with 'Novo Orçamento' and 'Novo Agendamento' buttons, and buttons for 'Hoje', '<', 'Mês' (highlighted), and '>'. The central part is a 7x7 grid representing a month's calendar. The days are labeled from 1 to 31. Specific dates are highlighted: '2' is in a blue box on September 9th; '1' is in a white box with a black border on September 11th; '5' is in a white box with a black border on September 16th; and '4' is in a white box with a black border on September 25th. At the bottom is a detailed agenda list for September 9th, titled 'Agendamentos - Dia 09/MM/xxxx'. It includes columns for 'Cliente', 'Hora do Agendamento', 'Previsão do Fim', and 'Valor Final'. Two entries are shown: 'Maria' at 'hh:mm' on 'dd/MM/yyyy hh:mm' with 'R\$ XXX,XX' and 'Mario' at 'hh:mm' on 'dd/MM/yyyy hh:mm' with 'R\$ X.XXX,XX'.

Cliente	Hora do Agendamento	Previsão do Fim	Valor Final
Maria	hh:mm	dd/MM/yyyy hh:mm	R\$ XXX,XX
Mario	hh:mm	dd/MM/yyyy hh:mm	R\$ X.XXX,XX

DV2 – Área de Parâmetros

Função

Agendamento

Autor Principal

Administrador

Fluxo de Eventos Principal

1. O sistema preenche o campo “Mês” com o mês atual
2. O sistema busca as informações no banco de dados
3. O sistema preenche o calendário com os dados já cadastrados
4. O sistema apresenta a tela (**DV1**)
5. O administrador clica em uma data (**A1)(A2)(A3)(A4**)
6. O sistema exibe detalhe do(s) agendamento(s) (**E1**)
7. O administrador pressiona em cima de um agendamento do detalhe (**A5**)
8. O sistema preenche o DV2 Função = “Exibir”
9. O sistema manda como parâmetro DV2 Agendamento os dados do agendamento
10. O sistema chama **UC010**
11. O caso de uso é encerrado

Fluxos Alternativos

A1: Botão “Novo Orçamento” pressionado

1. O sistema preenche o DV2 Função = “Novo”
2. O sistema chama **UC003**
3. O caso de uso é encerrado

A2: Botão “Novo Agendamento” pressionado

1. O sistema preenche o DV2 Função = “Novo”
2. O sistema chama **UC010**
3. O caso de uso é encerrado

A3: Botão “<” (seta para esquerda) ou “>” (seta para direita) pressionado

1. O sistema atualiza o nome no campo “Mês”
2. O sistema busca as informações no banco de dados
3. O sistema preenche o calendário
4. O sistema retorna ao fluxo principal

A4: Botão “Hoje” pressionado

1. O sistema busca as informações no banco de dados
2. O sistema preenche o calendário

3. O sistema exibe detalhe do(s) agendamento(s) da data atual (**E1**)
4. O sistema retorna ao fluxo principal

A5: Botão “Imprimir” pressionado

1. O sistema gera um documento em formato PDF com os detalhes exibidos
2. O sistema efetua a baixa do arquivo na máquina local
3. O caso de uso é encerrado

Fluxos de Exceção

E1: Administrador clica em uma data sem agendamentos

1. O sistema não exibe nenhum detalhe de agendamento
2. O sistema retorna ao fluxo principal

Regras de Negócio

R1: O sistema mostra (**DV1**) como tela inicial

UC002 – Listar Orçamento

Descrição

Este caso de uso serve para listar todos os orçamentos ordenados pela Data de Orçamento mais recente.

Data View

DV1 – Tela de listagem de orçamentos

DV2 – Área de Parâmetros

Função

Orcamento

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
2. O sistema preenche a tabela com os dados já cadastrados
3. O sistema apresenta a tela (**DV1**)
4. O administrador visualiza os resultados da tabela (**A1)(A2**)
5. O caso de uso é encerrado

Fluxos Alternativos

A1: Botão “Novo” pressionado

1. O sistema preenche o DV2 Função = “Novo”
2. O sistema chama **UC003**
3. O caso de uso é encerrado

A2: Orçamento pressionado

1. O sistema preenche o DV2 Função = “Exibir”
2. O sistema manda como parâmetro DV2 Orçamento os dados do orçamento
3. O sistema chama **UC003**
4. O caso de uso é encerrado

UC003 – Manter Orçamento

Descrição

Este caso de uso serve para manutenção de orçamentos.

Data View

DV1 – Aba de Dados da tela de manutenção de orçamentos

Orçamento

- Dados
- Serviços
- Produtos
- Peças

Cliente: Maria

Carro: KA

Total dos Produtos	Total das Peças	Total dos Serviços	Subtotal
R\$ XX,XX	R\$ XXX,XX	R\$ XXX,XX	R\$ XXX,XX

Desconto

Porcentagem	Valor
X%	R\$ X,XX

Valor Final: R\$ XXX,XX

Pagamento efetuado

DV2 – Área de Parâmetros

Função

Função-Cliente

Função-Serviço

Função-Agendamento

Função-Produto

Orçamento

DV3 – Aba de Serviços da tela de manutenção de orçamentos

DV4 – Aba de Produtos da tela de manutenção de orçamentos

DV5 – Aba de Peças da tela de manutenção de orçamentos

DV6 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

DV7 – FIGURA 12 – TELA DE CONFIRMAÇÃO DO BOTÃO EXCLUIR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro DV2 Função = “Novo” (**A1**)
 2. O sistema desabilita os botões ,  e 
 3. O sistema habilita os botões  e 
 4. O sistema habilita os campos de todas as abas
 5. O sistema busca informações no banco de dados
 6. O sistema preenche os campos de seleção com as informações
 7. O sistema apresenta a tela (**DV1**)
 8. O administrador preenche os dados referentes ao cliente (**A5(A11)**)

9. O administrador pressiona o botão  (A2)(A3)(A4)(E1)
10. O sistema salva as informações no banco de dados
11. O caso de uso é encerrado

Fluxos Alternativos

A1: Parâmetro “Exibir” recebido no DV2 Função

1. O sistema preenche as telas com os dados do Orçamento recebidos como parâmetro no DV2
2. O sistema desabilita os botões  e 
3. O sistema habilita os botões ,  e 
4. O sistema desabilita os campos de todas as abas
5. O sistema apresenta a tela (DV1)
6. O administrador visualiza os dados (A8)(A9)(A12)
7. O caso de uso é encerrado

A2: Aba de “Serviços” selecionada

1. O sistema apresenta a tela (DV3)
2. O administrador preenche os dados referente a serviços (A6)
3. O sistema retorna ao fluxo principal

A3: Aba de “Produtos” selecionada

1. O sistema apresenta a tela (DV4)
2. O administrador preenche os dados referente a produtos (A7)
3. O sistema retorna ao fluxo principal

A4: Aba de “Peças” selecionada

1. O sistema apresenta a tela (DV5)
2. O administrador preenche os dados referente a peças
3. O sistema busca “Valor Cobrado” do produto no banco de dados
4. O sistema calcula o “Total” pela “Qtde” vezes o “Valor do Produto”
5. O sistema preenche os campos “Valor Cobrado” e “Total” com os dados
6. O sistema retorna ao fluxo principal

A5: Botão “Novo” pressionado

1. O sistema preenche o DV2 Função-Cliente = “Novo”
2. O sistema chama **UC008** em uma nova janela do sistema
3. O sistema retorna ao fluxo principal

A6: Botão “Cadastrar Serviço” pressionado

1. O sistema preenche o DV2 Função-Serviço = “Novo”
2. O sistema chama **UC006** em uma nova janela do sistema
3. O sistema retorna ao fluxo principal

A7: Botão “Cadastrar Produto” pressionado

1. O sistema preenche o DV2 Função-Produto = “Novo”
2. O sistema chama **UC005** em uma nova janela do sistema
3. O sistema retorna ao fluxo principal

A8: Botão  pressionado

1. O sistema gera um documento em formato PDF com os detalhes exibidos
2. O sistema efetua a baixa do arquivo na máquina local
3. O caso de uso é encerrado

A9: Botão  pressionado

1. O sistema desabilita os botões ,  e 
2. O sistema habilita os botões  e 
3. O sistema habilita os campos de todas as abas
4. O administrador modifica os dados
5. O administrador pressiona o botão  **(A10)(E1)**
6. O sistema atualiza as informações no banco de dados
7. O caso de uso é reiniciado

A10: Botão  pressionado com DV2 Função = “Exibir”

1. O sistema apresenta a tela **(DV6)**
2. O administrador seleciona “Confirmar” **(A13)**
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A11: Botão  pressionado com DV2 Função = “Novo”

1. O sistema apresenta a tela **(DV6)**
2. O administrador seleciona “Confirmar” **(A13)**
3. O sistema retorna ao caso de uso que o chamou
4. O caso de uso é encerrado

A12: Botão  pressionado

1. O sistema apresenta a tela **(DV7)**
2. O administrador pressiona “Excluir”
3. O sistema exclui as informações no banco de dados **(E2)**
4. O sistema retorna ao **UC002**
5. O caso de uso é encerrado

A13: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

Fluxos de Exceção

E1: Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos
2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca os campos “Cliente” e “Carro”
4. O caso de uso é reiniciado

E2: Informações não podem ser excluídas

1. O sistema retorna a mensagem “Registro não pode ser excluído”
2. O caso de uso é reiniciado

UC004 – Editar Fornecedor

Descrição

Este caso de uso serve para edição dos fornecedores.

Data View

DV1 – Tela de edição de fornecedores

DV2 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
 2. O sistema preenche a tabela com os dados já cadastrados
 3. O sistema desabilita os botões  e 

4. O sistema habilita o botão 
5. O sistema desabilita os campos da tabela
6. O sistema apresenta a tela **(DV1)**
7. O administrador visualiza os dados **(A1)**
8. O caso de uso é encerrado

Fluxos Alternativos

A1: Administrador pressiona o botão 

1. O sistema desabilita botão 
2. O sistema habilita os botões  e 
3. O sistema habilita os campos da tabela
4. O administrador edita os dados da tabela
5. O sistema verifica se o conteúdo da coluna “Contato” é um e-mail válido de acordo com a regra de negócio **R1 (E2)**
6. O administrador pressiona o botão  **(A2)(E1)**
7. O sistema atualiza as informações no banco de dados
8. O caso de uso é encerrado

A2: Botão  pressionado

1. O sistema apresenta a tela **(DV2)**
2. O administrador seleciona “Confirmar” **(A3)**
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A3: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

Fluxos de Exceção

E1: Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos

2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca o campo “Nome”
4. O caso de uso é reiniciado

E2: E-mail inválido

1. O sistema destaca o campo “Contato” da linha inválida
2. O caso de uso retorna ao fluxo principal

Regras de Negócio

R1: O e-mail deve estar no formato “nome@dominio”

UC005 – Editar Produto

Descrição

Este caso de uso serve para edição dos produtos.

Data View

DV1 – Tela de edição de produtos

DV2 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
 2. O sistema preenche a tabela com os dados já cadastrados
 3. O sistema preenche a seleção de fornecedor

4. O sistema desabilita os botões  e 
5. O sistema habilita o botão 
6. O sistema desabilita os campos da tabela
7. O sistema apresenta a tela **(DV1)**
8. O administrador visualiza os dados **(A1)**
9. O caso de uso é encerrado

Fluxos Alternativos

A1: Administrador pressiona o botão 

1. O sistema desabilita botão 
2. O sistema habilita os botões  e 
3. O sistema habilita os campos
4. O administrador edita os dados da tabela
5. O administrador pressiona o botão  **(A2)(E1)**
6. O sistema atualiza as informações no banco de dados
7. O caso de uso é encerrado

A2: Botão  pressionado

1. O sistema apresenta a tela **(DV2)**
2. O administrador seleciona “Confirmar” **(A3)**
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A3: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

A4: Botão “Cadastrar Fornecedor” pressionado

1. O sistema preenche o DV2 Função-Fornecedor = “Novo”
2. O sistema chama **UC004** em uma nova janela do sistema

3. O sistema retorna ao fluxo principal

Fluxos de Exceção

E1: Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos
2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca os campos “Nome”, “Tipo”, “Valor Unitário” e “Valor Cobrado”
4. O caso de uso é reiniciado

UC006 – Editar Serviço

Descrição

Este caso de uso serve para edição dos serviços.

Data View

DV1 – Tela de edição de serviços

DV2 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
 2. O sistema preenche a tabela com os dados já cadastrados
 3. O sistema desabilita os botões  e 

4. O sistema habilita o botão 
5. O sistema desabilita os campos
6. O sistema apresenta a tela (**DV1**)
7. O administrador visualiza os dados (**A1**)
8. O caso de uso é encerrado

Fluxos Alternativos

A1: Administrador pressiona o botão 

1. O sistema desabilita botão 
2. O sistema habilita os botões  e 
3. O sistema habilita os campos
4. O administrador edita os dados da tabela 
5. O administrador pressiona o botão  (**A2)(E1**)
6. O sistema atualiza as informações no banco de dados
7. O caso de uso é encerrado

A2: Botão  pressionado

1. O sistema apresenta a tela (**DV2**)
2. O administrador seleciona “Confirmar” (**A3**)
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A3: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

Fluxos de Exceção

E1: Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos
2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca o campo “Nome”

4. O caso de uso é reiniciado

UC007 – Listar Cliente

Descrição

Este caso de uso serve para listar todos os clientes ordenados alfabeticamente.

Data View

DV1 – Tela de listagem de clientes

Nome	CPF	Contato Principal
Maria	XXX.XXX.XXX-XX	(XX)XXXXX-XXXX
Mario	XXX.XXX.XXX-XX	(XX)XXXXX-XXXX

DV2 – Área de Parâmetros

Função

Cliente

Autor Principal

Administrador

Fluxo de Eventos Principal

- O sistema busca as informações no banco de dados

2. O sistema preenche a tabela com os dados já cadastrados
3. O sistema apresenta a tela (**DV1**)
4. O administrador visualiza os resultados da tabela (**A1)(A2**)
5. O caso de uso é encerrado

Fluxos Alternativos

A1: Botão “Novo” pressionado

1. O sistema preenche o DV2 Função = “Novo”
2. O sistema chama **UC008**
3. O caso de uso é encerrado

A2: Cliente pressionado

1. O sistema preenche o DV2 Função = “Exibir”
2. O sistema manda como parâmetro DV2 Cliente os dados do cliente
3. O sistema chama **UC008**
4. O caso de uso é encerrado

UC008 – Manter Cliente

Descrição

Este caso de uso serve para manutenção de clientes.

Data View

DV1 – Aba de Dados da tela de manutenção de clientes

The screenshot shows a software application window titled "Cliente". The left sidebar has a "Clients" section expanded, showing "Agendamentos", "Listar Agendamentos", "Orçamentos", and "Clientes". Below that is a "Cadastros" section with "Fornecedores", "Produtos", and "Serviços". The main area is titled "Cliente" and contains tabs for "Dados", "Carros", "Orçamentos", and "Atendimentos". The "Dados" tab is selected, showing fields for "Nome" (Maria), "Contato Principal" (phone number (XX)XXXX-XXXX), "E-mail" (maria@mail.com), "CPF" (XXX.XXX.XXX-XX), and "CEP" (XX.XXX-XXX). There is also a large "Endereço" field containing "Rua XX, YY". On the right side of the main area are icons for delete, edit, and save.

DV2 – Área de Parâmetros

Função

Cliente

DV3 – Aba de Carros da tela de manutenção de clientes

DV4 – Aba de Orçamentos da tela de manutenção de clientes

DV5 – Aba de Atendimentos da tela de manutenção de clientes

The screenshot shows a software application window titled 'Cliente'. In the top right corner, there is a date and time display 'dd/MM/yyyy hh:mm'. On the left, a vertical sidebar lists several menu items: 'Agendamentos' (selected), 'Listar Agendamentos', 'Orçamentos', 'Clientes', 'Cadastros' (with a dropdown arrow), 'Fornecedores', 'Produtos', and 'Serviços'. The main content area is titled 'Atendimentos do Cliente' and contains a table with columns: 'Carro', 'Data/Hora do Agendamento', 'Data/Hora Fim do Serviço', 'Valor Final', and 'Pago'. The first row of the table has data: 'KA', 'dd/MM/yyyy hh:mm', 'dd/MM/yyyy hh:mm', 'R\$ XXX,XX', and a checked checkbox. Below the table, there are several empty rows for additional entries.

DV6 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

DV7 – FIGURA 12 – TELA DE CONFIRMAÇÃO DO BOTÃO EXCLUIR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro DV2 Função = “Novo” (A1)
2. O sistema desabilita os botões e
3. O sistema habilita os botões e
4. O sistema habilita os campos das abas “Dados” e “Carros”
5. O sistema busca informações no banco de dados
6. O sistema preenche os campos de seleção com as informações
7. O sistema apresenta a tela (DV1)
8. O administrador preenche os dados (A6)

9. O administrador seleciona a aba de “Carros”
10. O sistema apresenta a tela **(DV3)**
11. O administrador preenche os dados referente a carros
12. O administrador pressiona o botão  **(E1)(E2)**
13. O sistema verifica se o conteúdo do campo “E-mail” é um e-mail válido de acordo com a regra de negócio R1 **(E4)**
14. O sistema salva as informações no banco de dados
15. O caso de uso é encerrado

Fluxos Alternativos

A1: Parâmetro “Exibir” recebido no DV2 Função

1. O sistema preenche as telas com os dados do Cliente recebidos como parâmetro no DV2
2. O sistema desabilita os botões  e 
3. O sistema habilita os botões  e 
4. O sistema desabilita os campos de todas as abas
5. O sistema busca informações no banco de dados
6. O sistema preenche as tabelas das abas “Orçamentos” e “Atendimentos”
7. O sistema apresenta a tela **(DV1)**
8. O administrador visualiza os dados **(A2)(A3)(A4)**
9. O caso de uso é encerrado

A2: Aba de “Orçamentos” selecionada

1. O sistema apresenta a tela **(DV4)**
2. O administrador visualiza os dados
3. O sistema retorna ao fluxo principal

A3: Aba de “Atendimentos” selecionada

1. O sistema apresenta a tela **(DV5)**
2. O administrador visualiza os dados
3. O sistema retorna ao fluxo principal

A4: Botão  pressionado

1. O sistema desabilita os botões  e 
2. O sistema habilita os botões  e 
3. O sistema habilita os campos das abas “Dados” e “Carros”
4. O administrador modifica os dados
5. O administrador pressiona o botão  **(A5)(E1)(E2)**
6. O sistema verifica se o conteúdo do campo “E-mail” é um e-mail válido de acordo com a regra de negócio R1 **(E4)**
7. O sistema atualiza as informações no banco de dados
8. O caso de uso é reiniciado

A5: Botão  pressionado com DV2 Função = “Exibir”

1. O sistema apresenta a tela **(DV6)**
2. O administrador seleciona “Confirmar” **(A8)**
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A6: Botão  pressionado com DV2 Função = “Novo”

1. O sistema apresenta a tela **(DV6)**
2. O administrador seleciona “Confirmar” **(A8)**
3. O sistema retorna ao caso de uso que o chamou
4. O caso de uso é encerrado

A7: Botão  pressionado

1. O sistema apresenta a tela **(DV7)**
2. O administrador pressiona “Excluir” **(A8)**
3. O sistema exclui as informações no banco de dados **(E3)**
4. O sistema retorna ao **UC007**
5. O caso de uso é encerrado

A8: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

Fluxos de Exceção**E1:** Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos
2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca o campo “Nome”
4. O caso de uso é reiniciado

E2: Nenhum carro cadastrado

1. O sistema verifica que a tabela da aba “Carros” está vazia
2. O sistema retorna a mensagem “É preciso cadastrar ao menos um carro”
3. O caso de uso é reiniciado

E3: Informações não podem ser excluídas

1. O sistema retorna a mensagem “Registro não pode ser excluído”
2. O caso de uso é reiniciado

E4: E-mail inválido

1. O sistema destaca o campo “E-mail” da linha inválida
2. O caso de uso é reiniciado

Regras de Negócio**R1:** O e-mail deve estar no formato “nome@dominio”

UC009 – Listar Agendamento

Descrição

Este caso de uso serve para listar todos os agendamentos ordenados pela Data/Hora do Agendamento mais recente.

Data View

DV1 – Tela de listagem de agendamentos

DV2 – Área de Parâmetros

Função

Agendamento

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
2. O sistema preenche a tabela com os dados já cadastrados
3. O sistema apresenta a tela (**DV1**)
4. O administrador visualiza os resultados da tabela (**A1**)(**A2**)(**A3**)
5. O caso de uso é encerrado

Fluxos Alternativos

A1: Botão “Novo” pressionado

1. O sistema preenche o DV2 Função = “Novo”
2. O sistema chama **UC010**
3. O caso de uso é encerrado

A2: Agendamento pressionado

1. O sistema preenche o DV2 Função = “Exibir”
2. O sistema manda como parâmetro DV2 Agendamento os dados do agendamento
3. O sistema chama **UC010**
4. O caso de uso é encerrado

A3: Botão “Imprimir Agendamentos Diários” pressionado

1. O sistema gera um documento em formato PDF com os agendamentos finalizados do dia atual
2. O sistema efetua a baixa do arquivo na máquina local
3. O caso de uso é encerrado

UC010 – Manter Agendamento

Descrição

Este caso de uso serve para manutenção de agendamentos.

Data View

DV1 – Tela de manutenção de agendamentos

A interface DV1 é uma tela de manutenção de agendamentos. Ela possui uma barra lateral com menu, uma seção central para seleção de orçamento e uma seção para configuração de agendamento.

- Barra Lateral:** Contém links para "Agendamentos" (ativo), "Listar Agendamentos", "Orçamentos", "Clientes", "Cadastros" (ativo), "Fornecedores", "Produtos" e "Serviços".
- Seção Central (Selecionar Orçamento):**
 - Título: **Agendamento**
 - Botão: **Selecionar Orçamento**
 - Campos:
 - Cliente: Maria
 - Total dos Produtos: R\$ XX,XX
 - Total das Peças: R\$ XXX,XX
 - Total dos Serviços: R\$ XXX,XX
 - Subtotal: R\$ XXX,XX
 - Contato Principal: (DD)XXXXX-XXXX
 - Data do Orçamento: dd/MM/yyyy
 - Valor Final: R\$ XXX,XX
 - Botão: **Visualizar**
- Seção de Agendamento:**
 - Título: **Agendamento**
 - Campos:
 - Data do Agendamento: dd/MM/yyyy
 - Hora do Agendamento: hh:mm
 - Data Fim do Serviço: dd/MM/yyyy
 - Hora Fim do Serviço: hh:mm

DV2 – Área de Parâmetros

Função

Função-Orçamento

Agendamento

Orçamento

DV3 – FIGURA 11 – TELA DE CONFIRMAÇÃO DO BOTÃO CANCELAR

DV4 – FIGURA 12 – TELA DE CONFIRMAÇÃO DO BOTÃO EXCLUIR

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema recebe o parâmetro DV2 Função = “Novo” **(A1)(A8)**
2. O sistema desabilita os botões  ,  e 
3. O sistema habilita os botões  ,  e “Selecionar Orçamento”
4. O sistema desabilita os campos da área “Resumo do Orçamento”
5. O sistema habilita os campos da área “Agendamento”
6. O sistema apresenta a tela **(DV1)**
7. O administrador clica em “Selecionar Orçamento”
8. O sistema chama o **UC011**
9. O sistema recebe DV2 Orçamento com o orçamento selecionado no **UC011 (E2)**
10. O sistema preenche os dados da área “Resumo do Orçamento” com o orçamento recebido
11. O administrador preenche os dados da área “Agendamento”
12. O administrador pressiona o botão  **(A3)(A5)(E1)(E2)**
13. O sistema salva as informações no banco de dados
14. O caso de uso é encerrado

Fluxos Alternativos

A1: Parâmetro “Exibir” recebido no DV2 Função

1. O sistema preenche as telas com os dados do Agendamento recebidos como parâmetro no DV2
2. O sistema desabilita os botões  e 
3. O sistema habilita os botões  ,  e 
4. O sistema desabilita todos os campos
5. O sistema busca informações no banco de dados
6. O sistema preenche os campos
7. O sistema apresenta a tela **(DV1)**

8. O administrador visualiza os dados **(A3)(A9)**

9. O caso de uso é encerrado

A2: Botão  pressionado

1. O sistema desabilita os botões ,  e 
2. O sistema habilita os botões ,  e “Selecionar Orçamento”
3. O sistema habilita os campos da área “Agendamento”
4. O administrador modifica os dados
5. O administrador pressiona o botão  **(A4)(E1)(E2)**
6. O sistema atualiza as informações no banco de dados
7. O caso de uso é reiniciado

A3: Botão “Visualizar” pressionado

1. O sistema preenche o DV2 Função-Orçamento = “Exibir”
2. O sistema manda como parâmetro DV2 Orçamento os dados do orçamento
3. O sistema chama **UC003** em uma nova janela do sistema
4. O sistema retorna ao fluxo principal

A4: Botão  pressionado com DV2 Função = “Exibir”

1. O sistema apresenta a tela **(DV3)**
2. O administrador seleciona “Confirmar” **(A7)**
3. O sistema descarta as modificações
4. O caso de uso é reiniciado

A5: Botão  pressionado com DV2 Função = “Novo”

1. O sistema apresenta a tela **(DV3)**
2. O administrador seleciona “Confirmar” **(A7)**
3. O sistema retorna ao caso de uso que o chamou
4. O caso de uso é encerrado

A6: Botão  pressionado

1. O sistema apresenta a tela **(DV4)**
2. O administrador pressiona “Excluir” **(A7)**
3. O sistema exclui as informações no banco de dados **(E3)**
4. O sistema retorna ao caso de uso que o chamou
5. O caso de uso é encerrado

A7: Confirmação cancelada

1. O sistema fecha a tela de confirmação
2. O sistema retorna ao fluxo principal

A8: Recebe DV2 Função = “Novo” e DV2 Orçamento com parâmetro

1. O sistema preenche os dados da área “Resumo do Orçamento” com o orçamento recebido como parâmetro Orçamento no DV2
2. O sistema retorna ao fluxo principal

A9: Botão  pressionado

1. O sistema gera um documento em formato PDF com os detalhes de orçamento e agendamento (ordem de serviço)
2. O sistema efetua a baixa do arquivo na máquina local
3. O caso de uso é encerrado

Fluxos de Exceção

E1: Campos obrigatórios não preenchidos

1. O sistema verifica que os campos não foram preenchidos
2. O sistema retorna a mensagem “Informe os campos obrigatórios”
3. O sistema destaca os campos “Data do Agendamento” e “Hora do Agendamento”
4. O caso de uso é reiniciado

E2: Nenhum orçamento selecionado

1. O sistema verifica que nenhum orçamento foi selecionado
2. O sistema retorna a mensagem “É preciso selecionar um orçamento para agendar”
3. O caso de uso é reiniciado

E3: Informações não podem ser excluídas

1. O sistema retorna a mensagem “Registro não pode ser excluído”
2. O caso de uso é reiniciado

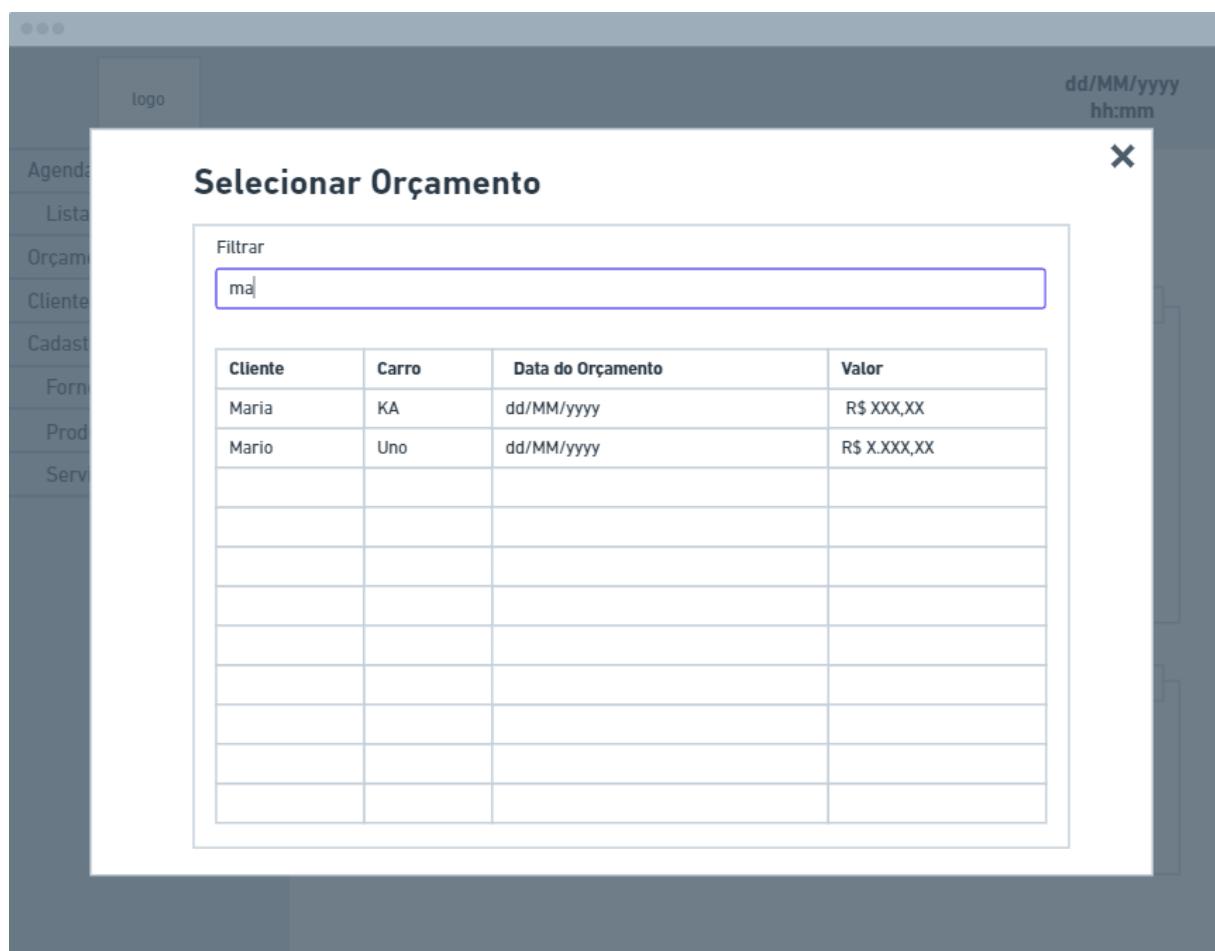
UC011 – Selecionar Orçamento

Descrição

Este caso de uso serve para selecionar um orçamento para tela de manutenção de agendamentos.

Data View

DV1 – Tela de seleção de orçamento



DV2 – Área de Parâmetros

Orçamento

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O sistema busca as informações no banco de dados
2. O sistema preenche a tabela com os dados já cadastrados
3. O sistema apresenta a tela (**DV1**)
4. O administrador seleciona um orçamento (**A1**)
5. O sistema manda como parâmetro DV2 Orçamento os dados do orçamento
6. O sistema retorna para o **UC010**
7. O caso de uso é encerrado

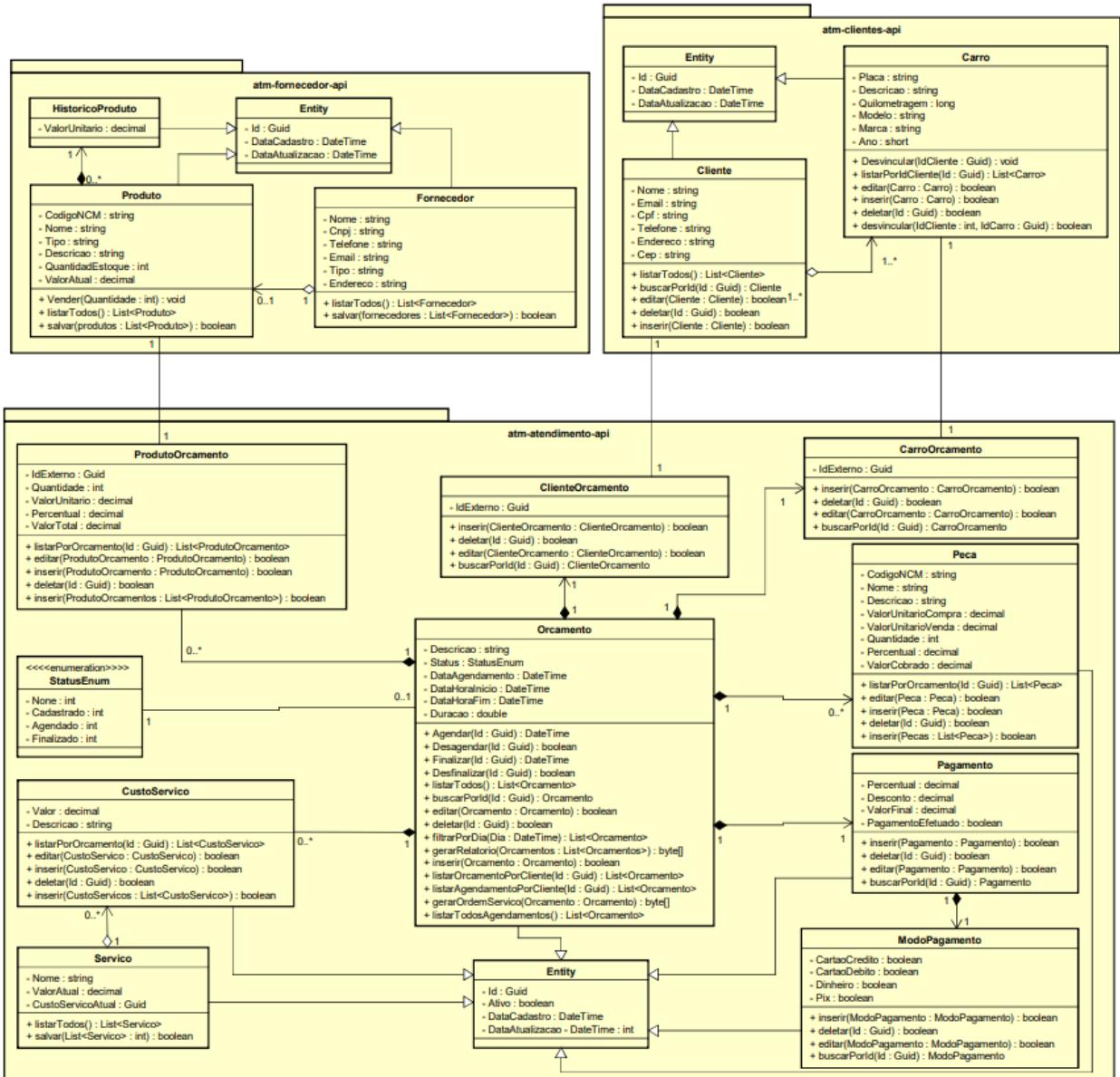
Fluxos Alternativos

A1: Administrador clica no botão “X”

1. O sistema retorna para o **UC010**
2. O caso de uso é encerrado

APÊNDICE 6 – DIAGRAMA DE CLASSES

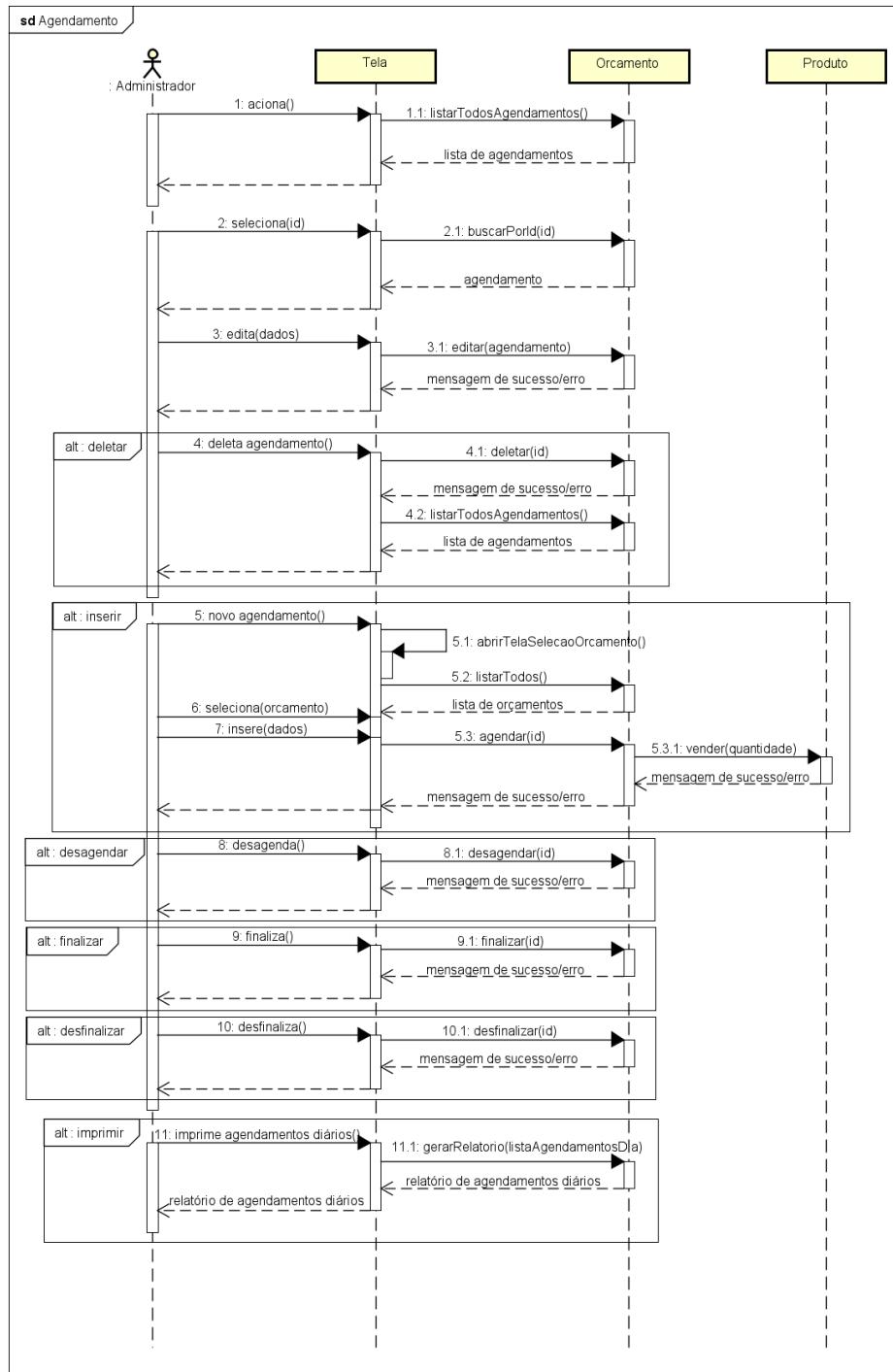
FIGURA 34 – DIAGRAMA DE CLASSES



FONTE: Os autores (2022)

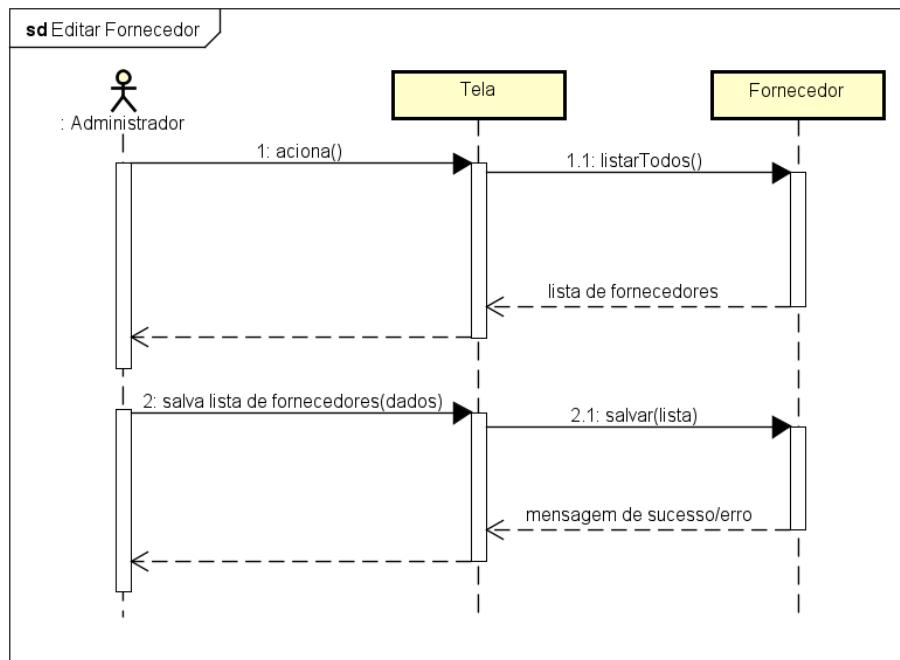
APÊNDICE 7 – DIAGRAMAS DE SEQUÊNCIA

FIGURA 35 - DIAGRAMA DE SEQUÊNCIA 001 – AGENDAMENTO



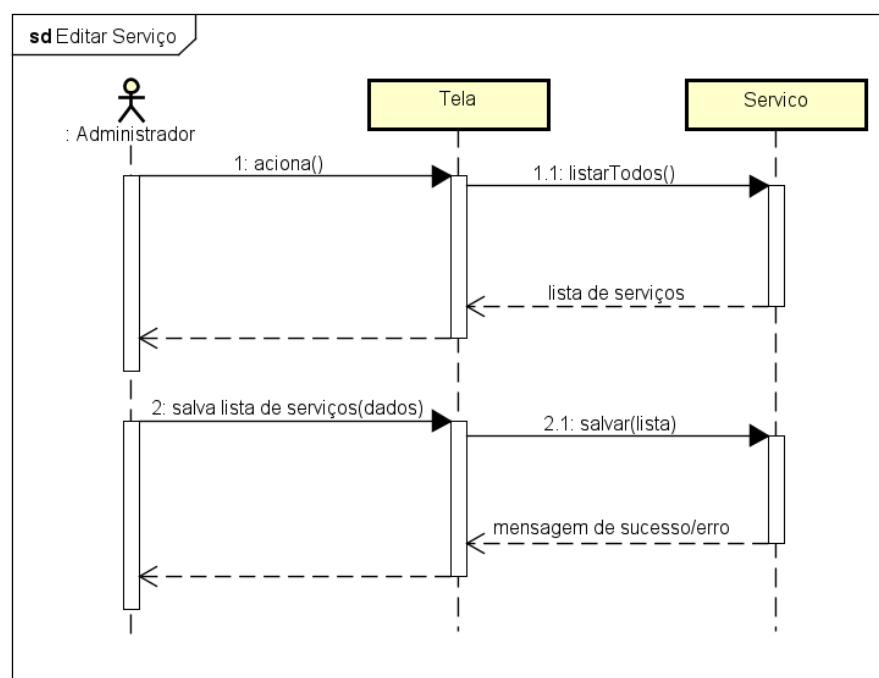
Fonte: Os autores (2022)

FIGURA 37 – DIAGRAMA DE SEQUÊNCIA 002 – EDITAR FORNECEDOR



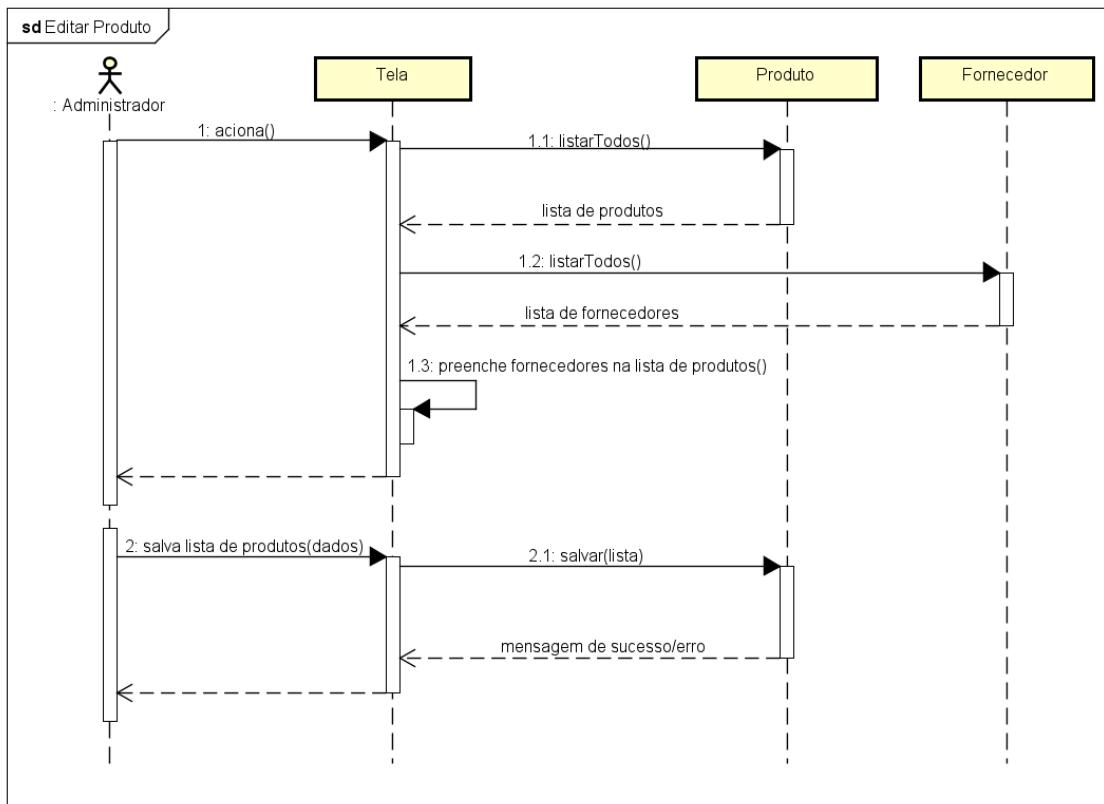
Fonte: Os autores (2022)

FIGURA 36 – DIAGRAMA DE SEQUÊNCIA 003 – EDITAR SERVIÇO



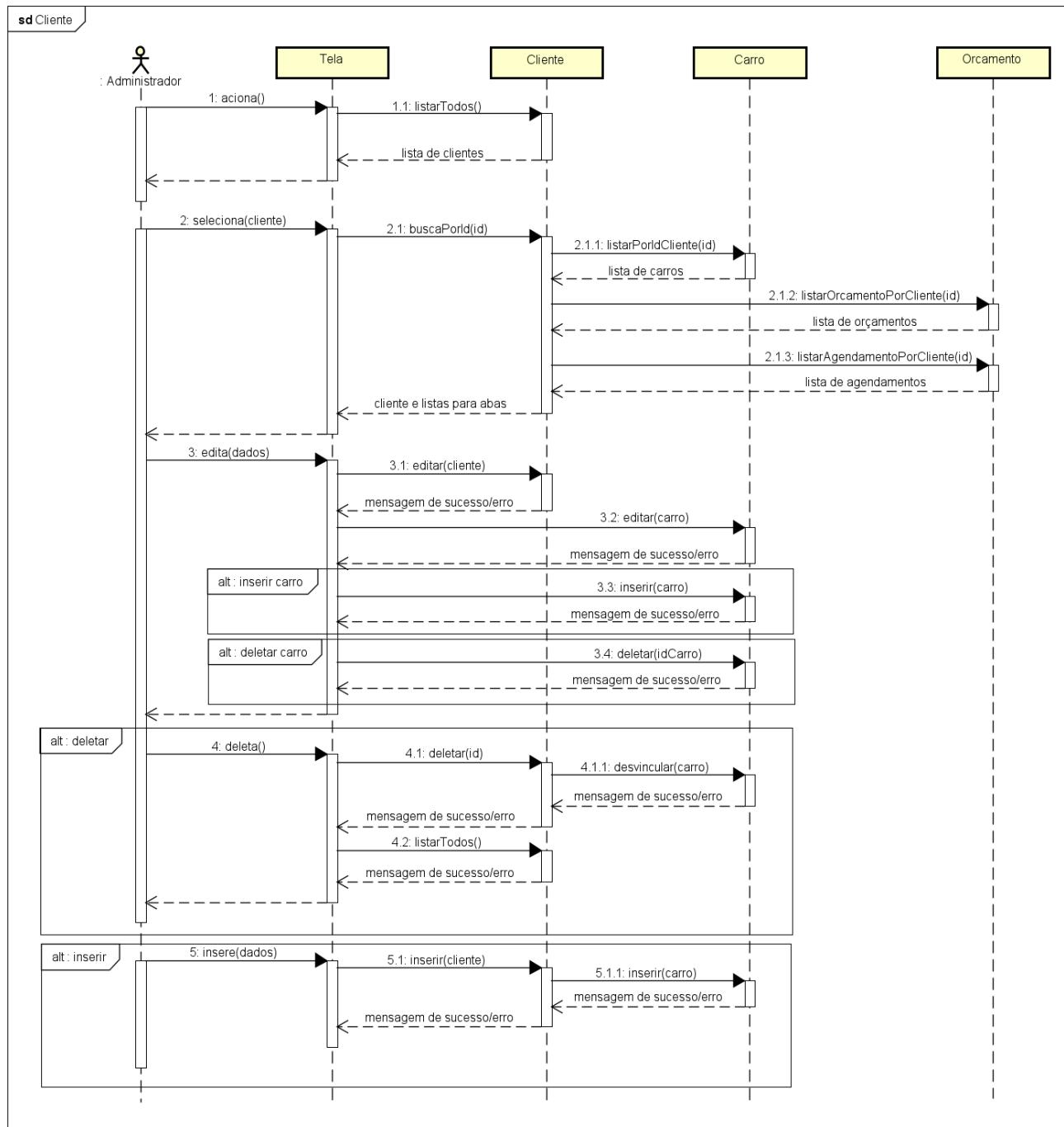
Fonte: Os autores (2022)

FIGURA 38 – DIAGRAMA DE SEQUÊNCIA 004 – EDITAR PRODUTO



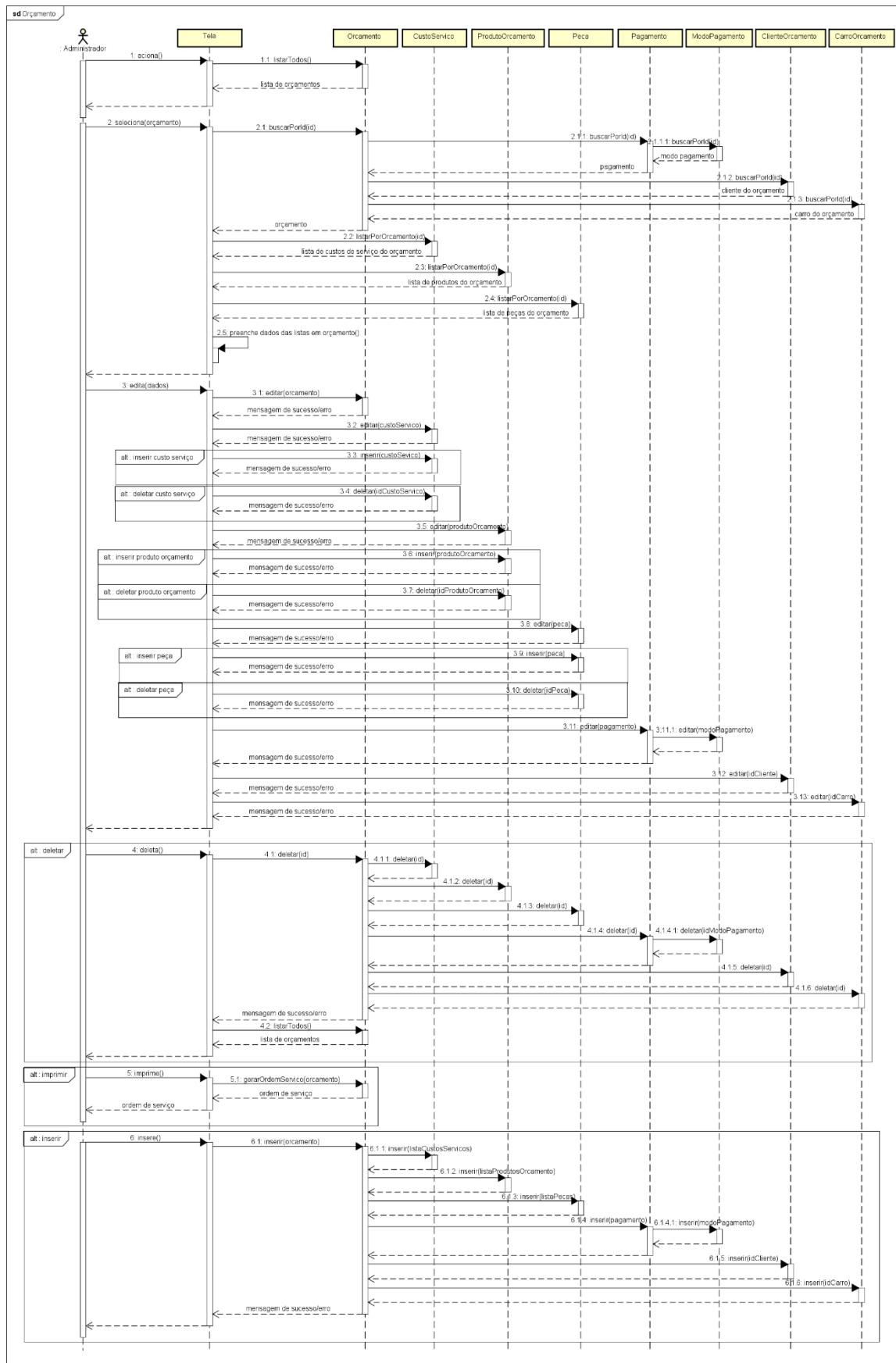
Fonte: Os autores (2022)

FIGURA 39 – DIAGRAMA DE SEQUÊNCIA 005 – CLIENTE



FONTE: Os autores (2022)

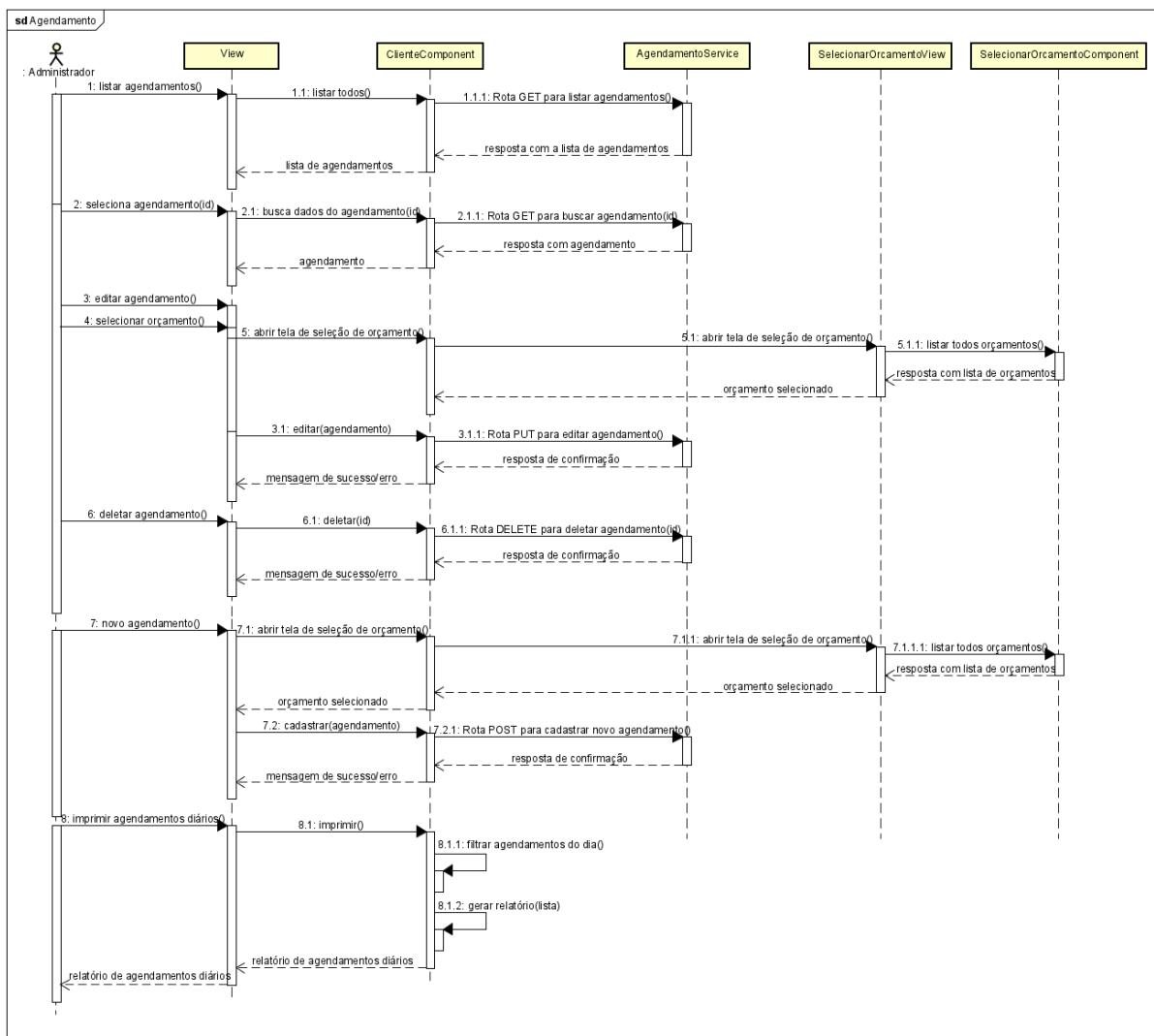
FIGURA 40 – DIAGRAMA DE SEQUÊNCIA 006 – ORÇAMENTO



FONTE: Os autores (2022)

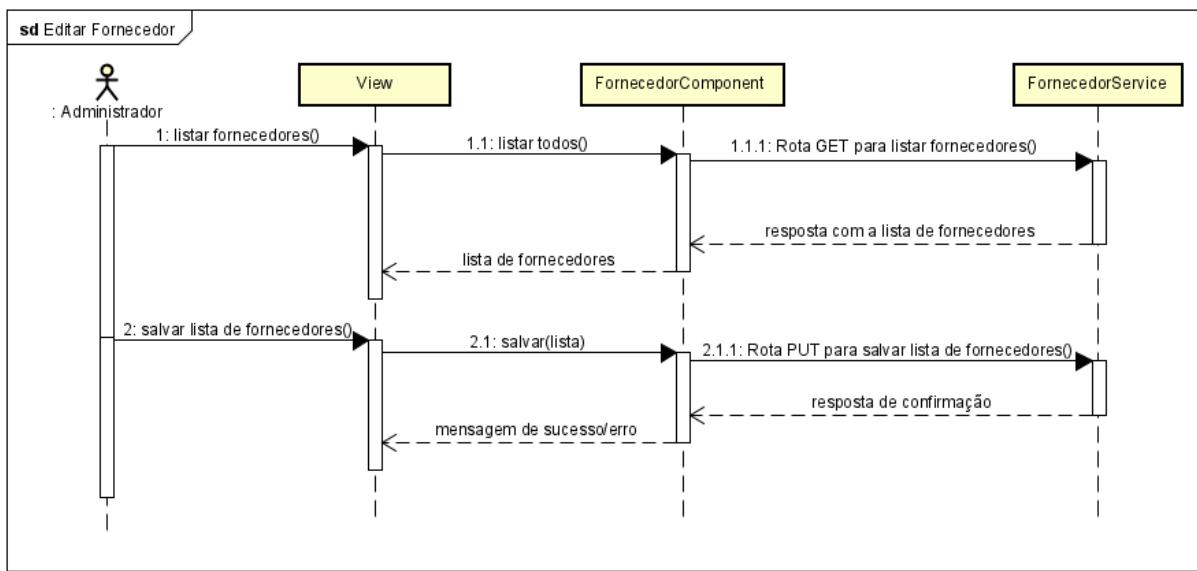
APÊNDICE 8 – DIAGRAMAS DE SEQUÊNCIA DE IMPLEMENTAÇÃO

FIGURA 41 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 001 – AGENDAMENTO



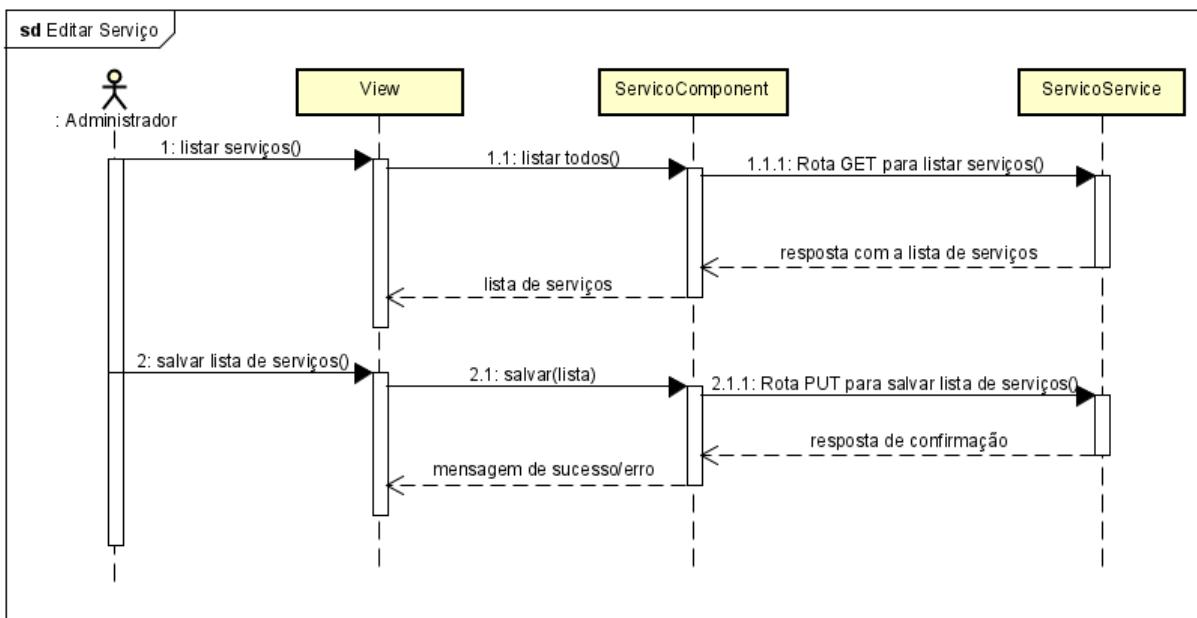
FONTE: Os autores (2021)

FIGURA 42 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 002 – EDITAR FORNECEDOR



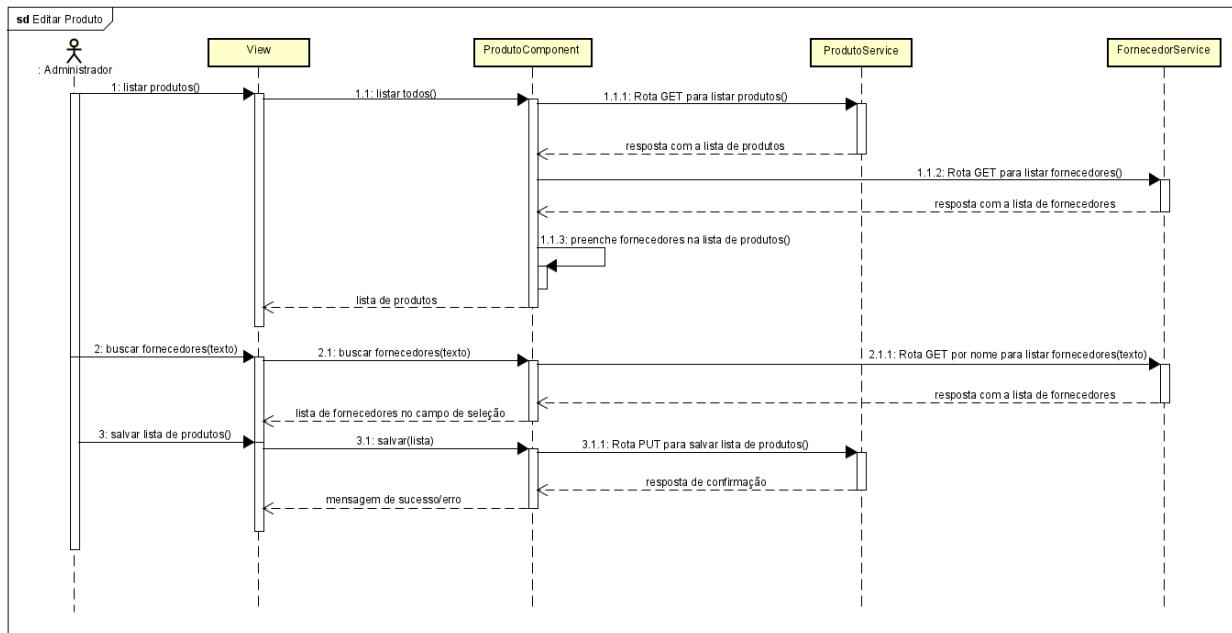
FONTE: Os autores (2021)

FIGURA 43 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 003 – EDITAR SERVIÇO



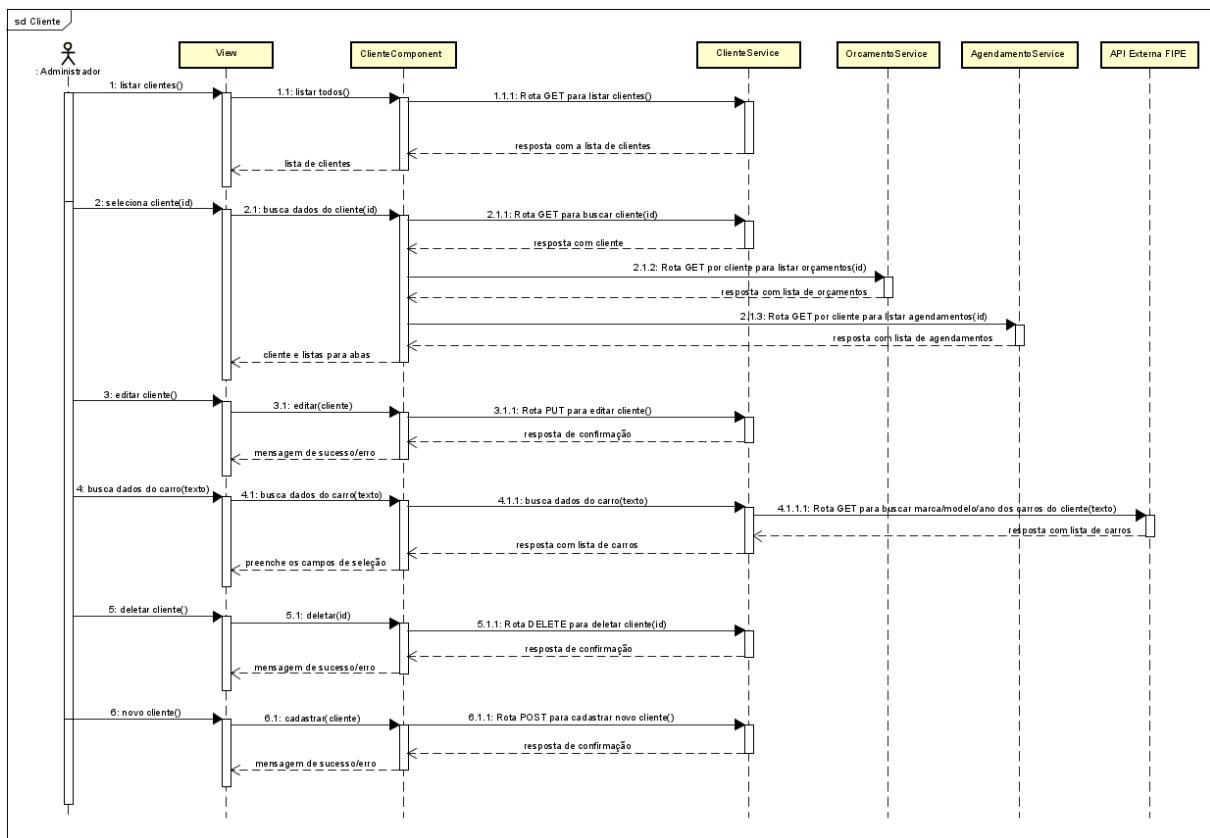
FONTE: Os autores (2021)

FIGURA 44 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 004 – EDITAR PRODUTO



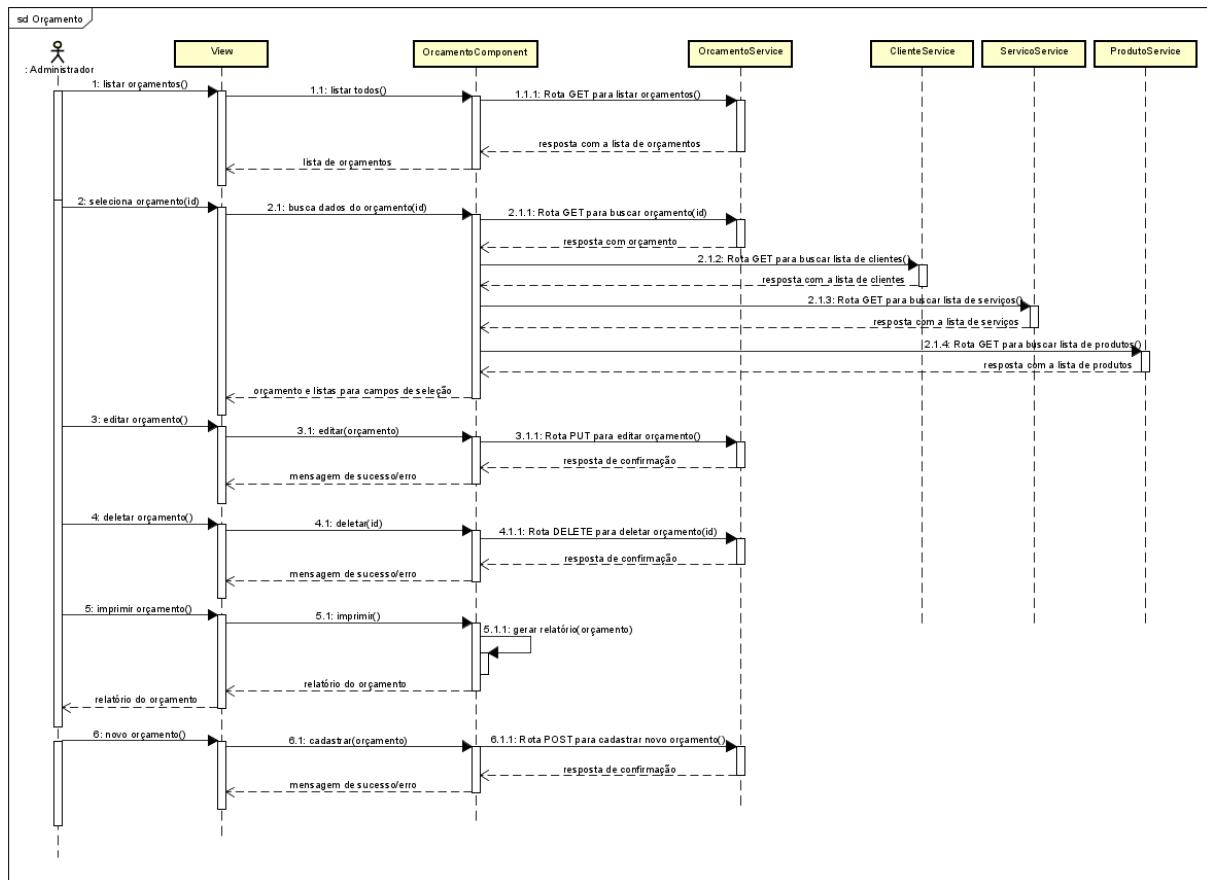
FONTE: Os autores (2021)

FIGURA 45 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 005 – CLIENTE



FONTE: Os autores (2021)

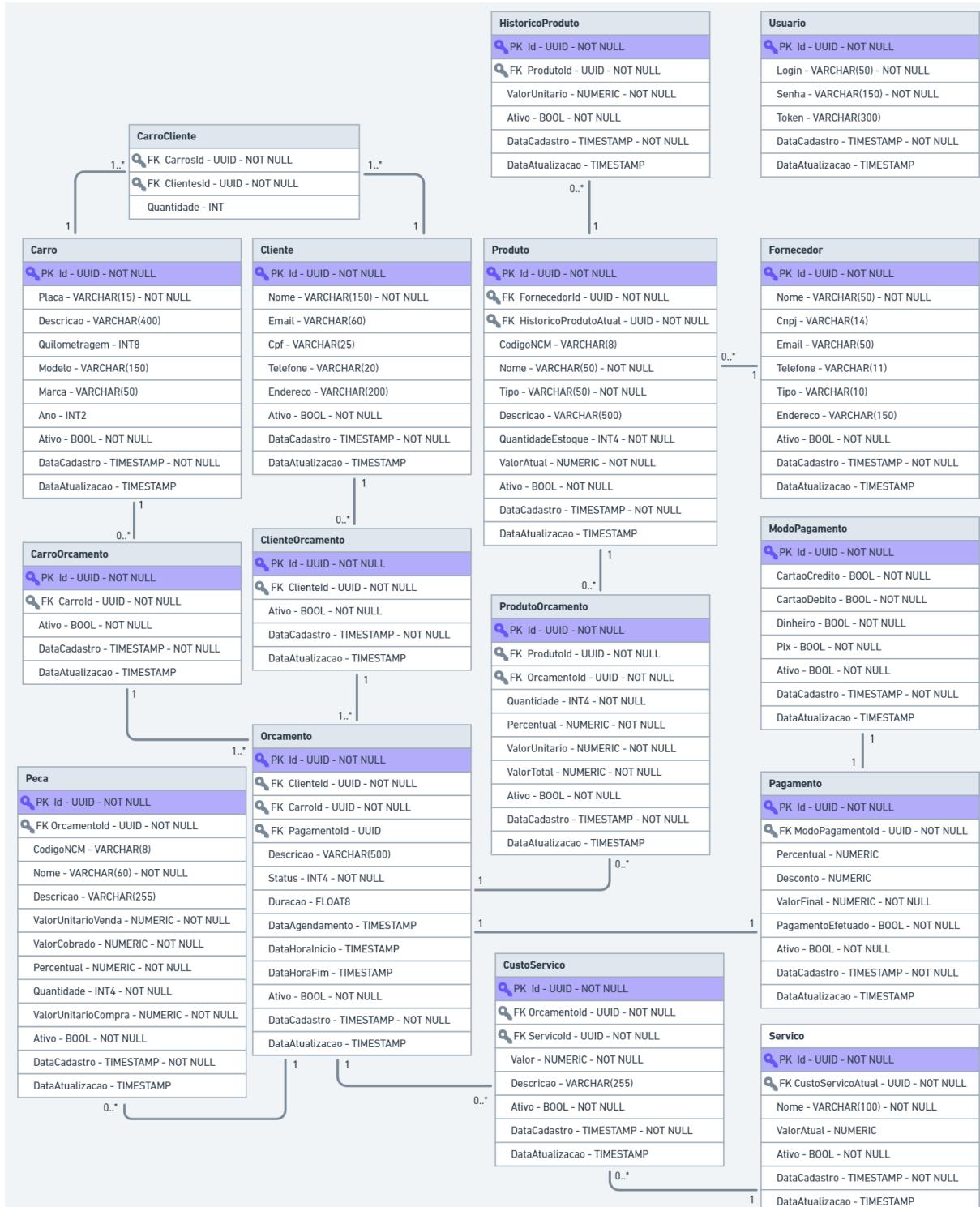
FIGURA 46 – DIAGRAMA DE SEQUÊNCIA DE IMPLEMENTAÇÃO 006 – ORÇAMENTO



FONTE: Os autores (2021)

APÊNDICE 9 – DIAGRAMA RELACIONAL

FIGURA 47 – DIAGRAMA RELACIONAL DO SISTEMA MARVIN



Fonte: Os autores (2022)

APÊNDICE 10 – DESCRIÇÃO DOS SPRINTS DA FASE DE MODELAGEM

- Sprint 1

O sprint 1 foi realizado entre 27/09/2021 e 04/10/2021. Houve um encontro com o cliente para entender suas necessidades e analisar a viabilidade de criar o trabalho de conclusão de curso em cima do pedido. Após o escopo do trabalho foi definido com base nas exigências fundamentais.

- Sprint 2

O sprint 2 foi realizado entre 04/10/2021 e 11/10/2021. A equipe levantou os requisitos preliminares do sistema e realizou o Diagrama de Classes.

- Sprint 3

O sprint 3 foi realizado entre 11/10/2021 e 18/10/2021. Os requisitos levantados foram validados com o cliente, e o protótipo (ou *mockup*) das telas do sistema foi efetuado.

- Sprint 4

O sprint 4 foi realizado entre 18/10/2021 e 25/10/2021. A equipe realizou uma pesquisa de sistemas similares ao proposto – sistemas gerenciadores de centros automotivos. Além disso, o cliente verificou a acurácia do Diagrama de Classes e do protótipo das telas.

- Sprint 5

O sprint 5 foi realizado entre 25/10/2021 e 08/11/2021. Correções apontadas pelo cliente foram integradas no Diagrama de Classes e nos protótipos das telas, e o Diagrama de Casos de Uso foi concluído.

- Sprint 6

O sprint 6 foi realizado entre 08/11/2021 e 15/11/2021. As Especificações de Casos de Uso foram feitas, bem como o Capítulo de Fundamentação Teórica. Foi dado início ao protótipo do sistema, começando pelo back-end da aplicação.

- Sprint 7

O sprint 7 foi realizado entre 15/11/2021 e 22/11/2021. O capítulo de Materiais e Métodos foi escrito, e o Diagrama de Sequência finalizado. O protótipo avança, tendo sua parte back-end terminada, e a front-end iniciada.

- Sprint 8

O sprint 8 foi realizado entre 22/11/2021 e 29/11/2021. O protótipo é concluído, com sua parte front-end pronta, e a integração entre as partes realizada e testada. O cliente tem sua primeira experiência com o sistema funcional e valida-o. O capítulo de Introdução é escrito.

- Sprint 9

O sprint 9 foi realizado entre 29/11/2021 e 06/12/2021. O protótipo é ajustado conforme requisitado pelo cliente, e os capítulos de Apresentação do Sistema e Considerações Finais são elaborados.

- Sprint 10

O sprint 10 foi realizado entre 06/12/2021 e 13/12/2021. Essa iteração foi reservada para correções necessárias no trabalho escrito.

APÊNDICE 11 – DESCRIÇÃO DOS SPRINTS DA FASE DE CONSTRUÇÃO

- Sprint 11

O sprint 11 foi realizado entre 31/01/2022 e 06/02/2022. Algumas classes foram refatoradas com base nos conselhos da banca do TCC 1, e foi dado início ao protótipo funcional da tela de orçamento e ao domínio de Cliente. Os domínios citados neste e nos próximos sprints têm como referência a Figura 5.

- Sprint 12

O sprint 12 foi realizado entre 07/02/2022 e 13/02/2022. O domínio de Cliente avança e sua interface é iniciada. Também ocorre a integração do WebService ViaCEP na interface de cliente, para facilitar o preenchimento do endereço. O protótipo é finalizado e validado com o cliente deste trabalho, o dono da empresa Automar.

- Sprint 13

O sprint 13 foi realizado entre 14/02/2022 e 20/02/2022. O domínio de cliente é finalizado. Na interface de cliente ocorre a integração com a API Externa da Tabela FIPE, que facilita o cadastro de carros ao buscar marcas, modelos e anos de veículos.

- Sprint 14

O sprint 14 foi realizado entre 21/01/2022 e 27/02/2022. O domínio de Atendimento inicia e a interface de Serviços é realizada.

- Sprint 15

O sprint 15 foi realizado entre 28/02/2022 e 06/03/2022. O domínio de Atendimento continua e a interface de Orçamento começa. O cliente valida o sistema incompleto e dá seu feedback.

- Sprint 16

O sprint 16 foi realizado entre 07/03/2022 e 13/03/2022. O domínio de Atendimento é concluído, enquanto a interface de Orçamento progride. Modificações no sistema são feitas com base no feedback fornecido pelo cliente.

- Sprint 17

O sprint 17 foi realizado entre 14/03/2022 e 20/03/2022. A interface de Orçamento e Cliente terminam. É dado início a interface de Agendamento e o cliente novamente testa o sistema inacabado.

- Sprint 18

O sprint 18 foi realizado entre 21/03/2022 e 27/03/2022. A interface de Agendamento avança e a de Seleção de Orçamento é realizada. Baseado no parecer do proprietário da empresa Automar, o sistema é alterado.

- Sprint 19

O sprint 19 foi realizado entre 28/03/2022 e 03/04/2022. A interface de Agendamento e Visualizar Agendamento.

- Sprint 20

O sprint 20 foi realizado entre 04/04/2022 e 10/04/2022. A interface de Agendamento e Visualizar Agendamento são finalizadas e o cliente valida o sistema. O cliente pede adição do campo de “Código NCM” em produto e peça.

- Sprint 21

O sprint 21 foi realizado entre 11/04/2022 e 17/04/2022. São feitas as modificações nos Domínios e Interfaces necessárias. Correções e revisões são feitas na monografia.

- Sprint 22

O sprint 22 foi realizado entre 18/04/2022 e 24/04/2022. O Domínio de autenticação é criado e finalizado, para validar todas as requisições para as APIs. A monografia continua pelo processo de correção e revisão.

- Sprint 23

O sprint 23 foi realizado entre 25/04/2022 e 01/05/2022. O front-end da aplicação é modificado para utilizar o token de autenticação em suas requisições para

as APIs. Também são feitas as modificações necessárias na monografia, bem como a última validação do sistema, agora completo, com o cliente.

- Sprint 24

O último sprint foi realizado entre 02/04/2022 e 06/04/2022. Pequenas modificações são realizadas no sistema e o mesmo é integrado através de script único. A monografia é finalizada.

APÊNDICE 12 – GUIA DE INSTALAÇÃO DO SISTEMA MARVIN

Instalação no sistema operacional Windows 10:

1. Abrir o prompt de comando ou o PowerShell como administrador.
2. Instalar o WSL³³ (Windows Subsystem for Linux) através do comando “wsl -- install”.
3. Reiniciar a máquina para efetivar a instalação do WSL.
4. Acessar o site do Docker e fazer o download do Docker Desktop³⁴.
5. Instalar o Docker Desktop.
6. Abrir a linha de comando do WSL (Ubuntu).
7. Navegar até a raiz da pasta setup do projeto atm-marvin-setup (ex.: cd /mnt/c/Users/{nome-do-usuario}/Desktop/[...]/atm-marvin-setup/setup).
8. Executar o comando “docker-compose pull” para baixar as imagens das aplicações no repositório local.

```
noh4nsen ➤ 19:43 ➤ .. /setup      main □ ➤ docker-compose pull
Pulling marvin-postgres-db    ... done
Pulling atm-autenticador-api ... done
Pulling atm-clientes-api     ... done
Pulling atm-fornecedor-api   ... done
Pulling atm-atendimento-api  ... done
Pulling atm-marvin-webapp   ... done
```

9. Conferir se as imagens foram baixadas corretamente através do comando “docker images”, em caso de erro executar o comando “docker-compose pull” novamente.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
noh4nsen/atm-clientes-api	latest	413ccb3b36e2	57 minutes ago	218MB
noh4nsen/atm-marvin-webapp	latest	681a97a5d1b2	44 hours ago	144MB
noh4nsen/atm-fornecedor-api	latest	d68c0514f113	2 days ago	218MB
noh4nsen/atm-atendimento-api	latest	8d10fe1b5aa0	3 days ago	217MB
noh4nsen/atm-autenticador-api	latest	82902ffc106b	13 days ago	217MB
postgres	latest	74b0c105737a	2 weeks ago	376MB

³³ WSL. Disponível em: <<https://docs.microsoft.com/pt-br/windows/wsl/install>> Acesso em: 13 dez. 2021.

³⁴ Docker Desktop. Disponível em: <<https://docs.docker.com/desktop/windows/install/>> Acesso em: 13 dez. 2021.

10. Executar o comando “*docker network create postgres-net*” para criar uma rede de comunicação interna do Docker na qual os microsserviços se comunicarão (A criação da rede pode ser conferida através do comando “*docker network ls*”).

```
noh4nsen ➤ 19:44 ➤ .. /setup main ➤ docker network create postgres-net
1a3004f35c32730a046176fbde0b74b109a15af761f68256ab9456be28b3739b
```

noh4nsen ➤ 19:55 ➤ ~ docker network ls			
NETWORK ID	NAME	DRIVER	SCOPE
b28ec2900ac1	bridge	bridge	local
9212c85b0d0c	host	host	local
03f79e55ba72	none	null	local
b6bd99520552	postgres-net	bridge	local

11. Executar o comando “*docker-compose up -d*” para iniciar os contêineres.

```
noh4nsen ➤ 19:54 ➤ .. /setup main ➤ docker-compose up -d
Creating marvin-postgres-db ... done
Creating atm-autenticador-api ... done
Creating atm-fornecedor-api ... done
Creating atm-clientes-api ... done
Creating atm-atendimento-api ... done
Creating marvin ... done
```

12. O correto funcionamento dos contêineres pode ser verificado através do comando “*docker ps*”.

noh4nsen ➤ 19:55 ➤ ~ docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7feee5a8a5d	noh4nsen/atm-marvin-webapp:latest	"/docker-entrypoint..."	35 seconds ago	Up 34 seconds	0.0.0.0:80→80/tcp	marvin
3707356ebbe4	noh4nsen/atm-atendimento-api:latest	"dotnet Atm.Atendimento..."	37 seconds ago	Up 35 seconds	0.0.0.0:5678→80/tcp, 0.0.0.0:5679→443/tcp	atm-atendimento-api
6023f8202b3e	noh4nsen/atm-fornecedor-api:latest	"dotnet Atm.Fornecedor..."	39 seconds ago	Up 37 seconds	0.0.0.0:5675→443/tcp	atm-fornecedor-api
ef7865a03b70	noh4nsen/atm-clientes-api:latest	"dotnet Atm.Clientes..."	39 seconds ago	Up 37 seconds	0.0.0.0:5676→80/tcp, 0.0.0.0:5677→443/tcp	atm-clientes-api
b28907d64f1e	noh4nsen/atm-autenticador-api:latest	"dotnet Atm.Autentic..."	41 seconds ago	Up 39 seconds	0.0.0.0:5680→80/tcp, 0.0.0.0:5681→443/tcp	atm-autenticador-api
df2f81528fa1c	postgres	"docker-entrypoint.s..."	42 seconds ago	Up 40 seconds	0.0.0.0:5432→5432/tcp	marvin-postgres-db

13. Efetive a importação dos certificados de desenvolvimento na pasta */certificates/https/* (senha: marvin-atm).
14. Acesse alguma das APIs (ex.: API de Clientes em <https://localhost:5677>) e desative os avisos de segurança para aquele domínio. Dependendo do navegador utilizado, é preciso acessar todas as portas (5675, 5677, 5679 e 5681).
15. Após a instalação de todos os projetos, o sistema estará disponível através do navegador no URI <http://localhost>.