
Fatec Araras “Antônio Brambilla”

Documentação de Software
Lava Rápido
Projeto Integrador - 2º Semestre - DSM

<i>Integrantes</i>
<i>Lucas Eduardo Rosolem</i>
<i>Bruno Eduardo Rodrigues</i>
<i>Renan Augusto Eugenio Marques</i>
<i>Willian Adriano Dias Paulino</i>
<i>Eliabe Leme Vicente da Silva</i>

Anexo II- Cronograma Geral do Projeto

Integrantes	Descrição	Data de Início	Data de Término
Lucas	Criação do repositório no Github	04/03/2025	04/03/2025
Grupo	Definição do projeto (Votação)	17/03/2025	18/03/2025
Grupo	“Quadro de Trabalho”, Metodologia Ágil (Scrum) Ferramenta (Trello)	19/03/2025	Utilização até o final do projeto!
Eliabe / Renan	Prototipação (Figma)	13/04/2025	04/05/2025
Lucas	Modelagem de Banco de dados	03/04/2025	31/05/2025
Lucas	Criação do banco de dados mysql e tabelas	06/04/2025	31/05/2025
Lucas	Diagramas UML	06/04/2025	08/05/2025
Willian	Levantamento de Requisitos	21/03/2025	21/03/2025
Bruno	Introdução	13/04/2025	06/04/2025
Bruno	Escopo	13/04/2025	06/04/2025
Lucas Renan	Codificação	03/05/2025	24/06/2025
Willian Bruno	Requisitos funcionais e não funcionais	27/04/2025	28/05/2025
Lucas	Criação de objetos no banco de dados	15/06/2025	15/06/2025
Grupo	Documentação	18/03/2025	01/07/2025
Lucas Renan	Hospedagem do sistema	27/06/2025	01/07/2025

Sumário

0.0 Cronograma Geral do Projeto

1. INTRODUÇÃO

- 1.1 Apresentação da Empresa
- 1.2 Missão, Visão e Valores
- 1.3 Repositório

2. ESCOPO DO SOFTWARE

- 2.1 Objetivo do Projeto
- 2.2 Levantamento de Requisitos (LR)
- 2.3 Requisitos funcionais (RF)
- 2.4 Requisitos Não Funcionais (RNF)

3. ESPECIFICAÇÃO DE SOFTWARE

- 3.1 Diagramas UML
 - 3.1.1 Diagrama de Caso de Uso
 - 3.1.2 Diagrama de Classe
 - 3.1.3 Diagrama de Sequência
- 3.2 Modelagem de Banco de Dados
 - 3.2.1 Modelo Conceitual
 - 3.2.2 Modelo Lógico
- 3.3 Recursos

4. DESENVOLVIMENTOS (EXECUÇÃO)

- 4.1 Banco de Dados
- 4.2 Prototipação
 - 4.2.1 Descrição dos Protótipos + Interface do Usuário
 - 4.2.2 Validação dos Protótipos + Feedbacks
- 4.3 Codificação

5. TESTES E QUALIDADE

- 5.1 Estratégia de Testes: Descrever a estratégia (método) de testes adotada
- 5.2 Resultados dos Testes: Apresentar os resultados dos testes realizados
- 5.3 Garantia da Qualidade: Descrever as práticas adotadas para garantia da qualidade
- 5.4 Requisitos Mínimos de Hardware e Software

6. CONTRATO PARA DESENVOLVIMENTO DE SOFTWARE

7. PLANO DE EVOLUÇÃO

8. REFERÊNCIAS

- Anexo I - Diário de bordo
- Anexo II – Cronograma efetivo
- Anexo III – Contrato

1. INTRODUÇÃO

1.1 Apresentação da empresa

Este documento descreve a especificação do projeto de uma aplicação web, para a gestão de agendamentos de veículos em lava-jatos. A solução foi idealizada como parte do Projeto Integrador (PI) do curso de **Desenvolvimento de Sistemas Multiplataformas** da **Fatec Araras**, desenvolvida pelo grupo **Web Lab**, e tem como objetivo principal facilitar o atendimento ao cliente, otimizar o tempo dos prestadores de serviço e modernizar o processo de contratação de serviços oferecidos.

A aplicação será implementada para atender às necessidades do cliente real **Embelezamento Automotivo**, um lava-rápido que busca oferecer serviços de lavagem e cuidados automotivos com qualidade, agilidade e foco na satisfação do cliente. O sistema proposto permitirá que os usuários realizem agendamentos online, escolham o tipo de serviço desejado, visualizem horários disponíveis, tudo por meio de uma interface intuitiva e responsiva.

Com esta aplicação, o **Embelezamento Automotivo** poderá melhorar o gerenciamento da sua agenda, reduzir a taxa de desistências e garantir maior comodidade aos seus clientes. O projeto também visa aplicar os conhecimentos adquiridos ao longo do curso, integrando áreas como análise de requisitos, desenvolvimento web, banco de dados e segurança da informação, promovendo uma experiência real de desenvolvimento de software em equipe.

1.2 Missão visão e valores

Missão

Desenvolver soluções digitais inovadoras, funcionais e de qualidade, que atendam às necessidades reais dos clientes, promovendo aprendizado prático, colaboração em equipe e aplicação dos conhecimentos adquiridos ao longo do curso de Desenvolvimento de Sistemas Multiplataformas da Fatec Araras.

Visão

Ser reconhecido como um grupo de referência em desenvolvimento de sistemas no ambiente acadêmico, destacando-se pela competência técnica, comprometimento com a entrega de soluções eficientes e pela busca contínua por inovação e excelência.

Valores

- **Comprometimento:** Cumprir prazos, responsabilidades e entregas com seriedade e dedicação.
- **Colaboração:** Trabalhar em equipe de forma respeitosa, solidária e produtiva.
- **Inovação:** Buscar constantemente novas ideias, ferramentas e soluções criativas.
- **Qualidade:** Entregar produtos funcionais, bem estruturados e focados na experiência do usuário.
- **Aprendizado contínuo:** Valorizar o desenvolvimento técnico e pessoal de cada integrante.
- **Ética:** Agir com honestidade, responsabilidade e respeito em todas as etapas do projeto.

1.3 Repositório

Todo o projeto será disponibilizado no repositório do Github, no seguinte endereço eletrônico: [GitHub - Lucas-Ed/Pi-2-semester: Sistema de Lava Rápido](#), o projeto, apesar de ser público detém a seguinte licença de software Creative Commons Attribution-NonCommercial 4.0 (CC BY-NC 4.0).

2. ESCOPO

O escopo do projeto inclui o desenvolvimento com as seguintes limitações:

O escopo deste projeto contempla o desenvolvimento de uma aplicação web voltada ao agendamento de serviços de lavagem automotiva para o cliente **Embelezamento Automotivo**. A aplicação terá como foco principal a automatização do processo de agendamento, visando oferecer praticidade, agilidade e melhor controle para o cliente e para o negócio. Dentre as funcionalidades previstas, destacam-se:

- Cadastro e login de clientes;
- Visualização dos serviços disponíveis e seus valores;
- Agendamento de serviços com escolha de data e horário;
- Painel administrativo para controle de agendamentos e visualização de clientes;
- Interface **web responsiva**, adaptável a diferentes dispositivos (computadores, tablets e smartphones).

Apesar do escopo ser abrangente, o projeto será desenvolvido com algumas **limitações**, determinadas para garantir sua execução dentro do cronograma, dos recursos disponíveis e dos objetivos pedagógicos do curso:

- O sistema será restrito ao ambiente web, **sem o desenvolvimento de aplicativos mobile nativos**;
- O foco principal será em funcionalidades essenciais, **sem a inclusão de funcionalidades avançadas como programas de fidelidade, emissão de nota fiscal ou relatórios personalizados em tempo real**;

- **A infraestrutura de produção poderá ser implementada em servidores de baixo custo**, considerando a natureza acadêmica do projeto;
- O sistema será projetado com atenção à segurança, porém, **recursos de segurança avançada (como autenticação multifator ou criptografia de ponta a ponta) não estão previstos nesta versão inicial**.

As decisões sobre o escopo e suas limitações foram tomadas com base na viabilidade técnica, no tempo disponível e nas prioridades definidas junto ao cliente. Mesmo com essas limitações, o projeto visa entregar um produto mínimo viável funcional, confiável e alinhado às necessidades reais da empresa **Embelezamento Automotivo**.

2.1 Objetivo do projeto

O objetivo deste projeto é desenvolver uma aplicação web para o cliente **Embelezamento Automotivo**, com foco na modernização e automatização dos processos de **agendamento de serviços** de lavagens automotivas. A proposta visa oferecer uma solução digital intuitiva e eficiente, que melhore a experiência dos clientes e otimize a gestão do negócio.

Com essa aplicação, busca-se facilitar o controle de horários disponíveis, evitar conflitos de agenda, reduzir o tempo de espera e proporcionar maior comodidade aos usuários, que poderão agendar serviços e acompanhar o status do agendamento pela internet. Para o cliente, o sistema representa uma oportunidade de profissionalizar a operação, fidelizar clientes e ganhar competitividade no mercado.

Além disso, o projeto tem como objetivo integrar os conhecimentos adquiridos pelos integrantes do grupo **Web Lab** durante o curso de **Desenvolvimento de Sistemas Multiplataformas** da **Fatec Araras**, permitindo a aplicação prática de conceitos de desenvolvimento web, banco de dados, usabilidade e gestão de projetos em um cenário real de desenvolvimento de software.

2.2 Levantamento de requisitos

O levantamento de requisitos foi feito através de uma breve reunião com o proprietário do Lava-Rápido, para podermos identificar as principais necessidades e expectativas. Foi analisado que a maior dificuldade que o proprietário enfrenta no dia a dia é na parte de agendamento de clientes, onde o cliente muitas vezes “agendam” para realizar um serviço complexo e no dia/horário não cumpre com o combinado e não leva o veículo, acarretando em perdas de tempo(dinheiro) e de outros serviços.

Nosso cliente (proprietário do lava rápido) espera do nosso projeto uma nova ferramenta para o estabelecimento onde tenha maior agilidade no atendimento e organização de novos clientes.

2.3 Requisitos funcionais (RF)

Requisitos Funcionais Administrador	
Definição:	<u>Login</u>
Ator:	Administrador
Prioridade:	Essencial.
RNF Associados:	RNF - 001 / RNF - 003
Código:	RF - 001
Descrição:	O sistema deve permitir que o administrador acesse o sistema por meio de autenticação com e-mail e senha válidos e o administrador insira e-mail e senha para login.

Definição	<u>Cancelar Agendamento</u>
Ator:	Administrador
Prioridade:	Essencial.
RNF Associados:	RNF - 004 / RNF - 005
Código:	RF - 002
Descrição:	O sistema deve permitir que o administrador cancele um agendamento existente.

Definição	<u>Consultar Agendamentos</u>
Ator:	Administrador
Prioridade:	Essencial.
RNF Associados:	RNF - 006 / RNF - 007
Código:	RF - 003
Descrição:	O sistema deve permitir que o administrador consulte a lista de agendamentos.

Definição	<u>Consultar Clientes</u>
Ator:	Administrador
Prioridade:	Essencial.
RNF Associados:	RNF - 008 / RNF - 009
Código:	RF - 004
Descrição:	O sistema deve permitir que o administrador consulte informações dos clientes cadastrados.

Requisitos Funcionais Cliente	
Definição	Cadastro.
Ator:	Cliente.
Prioridade:	Essencial.
RNF Associados:	RNF - 001 / RNF - 002 / RNF - 003.
Código:	RF - 005
Descrição:	O sistema deve permitir que o cliente acesse sua conta por meio de autenticação com e-mail e senha válidos.

Definição	Login.
Ator:	Cliente.
Prioridade:	Essencial.
RNF Associados:	RNF - 008 / RNF - 009.
Código:	RF - 006
Descrição:	O sistema deve permitir que o cliente insira e-mail e senha para login.

Definição	<i>Recuperação de senha.</i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 002 / RNF - 010.</i>
Código:	<i>RF - 007</i>
Descrição:	<i>O sistema deve permitir que o cliente solicite a recuperação de senha por meio do envio de código ao e-mail cadastrado.</i>

Definição	<i>Agendar serviço.</i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 006 / RNF - 007.</i>
Código:	<i>RF - 008</i>
Descrição:	<i>O sistema deve permitir que o cliente agende um serviço, selecionando data, horário, tipo de serviço e veículo.</i>

Definição	<i>Cadastrar veículo.</i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 008 / RNF - 009.</i>
Código:	<i>RF - 009</i>
Descrição:	<i>O sistema deve permitir que o cliente cadastre um ou mais veículos informando dados como placa, modelo, cor e tipo.</i>

Definição	<i>Remover veículos.</i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 007 / RNF - 011.</i>
Código:	<i>RF - 010</i>
Descrição:	<i>O sistema deve permitir que o cliente visualize seus veículos cadastrados.</i>

Definição	<i>Cancelar agendamento.</i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 004/RNF - 005</i>
Código:	<i>RF - 011</i>
Descrição:	<i>O sistema deve permitir que o cliente visualize os agendamentos ativos.</i>

Definição	<i><u>Consultar agendamento.</u></i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 007 / RNF - 009.</i>
Código:	<i>RF - 012</i>
Descrição:	<i>O sistema deve permitir que o cliente visualize a lista de agendamentos futuros.</i>

Definição	<i><u>Logout.</u></i>
Ator:	<i>Cliente.</i>
Prioridade:	<i>Essencial.</i>
RNF Associados:	<i>RNF - 010.</i>
Código:	<i>RF - 013</i>
Descrição:	<i>O sistema deve permitir que o cliente encerre a sessão a qualquer momento.</i>

2.4 Requisitos não funcionais (RNF)

Requisitos Não Funcionais	
Código	Descrição
RNF - 001	O tempo de resposta do login não deve ultrapassar 5 segundos.
RNF - 002	As senhas e CPF's devem ser armazenadas de forma criptografada.
RNF - 003	A sessão deve expirar após 1800 segundos equivalentes a 30 minutos de inatividade.
RNF - 004	O cancelamento deve ser refletido para o cliente em tempo real.
RNF - 005	Apenas administradores autenticados devem poder realizar o cancelamento.
RNF - 006	Os resultados da consulta devem ser exibidos em até 5 segundos.
RNF - 007	A interface deve ser responsiva e compatível com diferentes dispositivos.
RNF - 008	Os dados dos clientes devem estar protegidos conforme a LGPD.
RNF - 009	A confidencialidade e integridade das informações devem ser garantidas.
RNF - 010	O código de recuperação de senha deve expirar após 2 horas por segurança.
RNF - 011	O sistema deve impedir a exclusão de veículos com agendamentos ativos.
RNF - 012	Após logout, o sistema deve invalidar imediatamente a sessão atual.
RNF - 013	A remoção deve ser imediata e confirmada ao usuário.

3 ESPECIFICAÇÃO DE SOFTWARE

3.1 .Diagramas UML

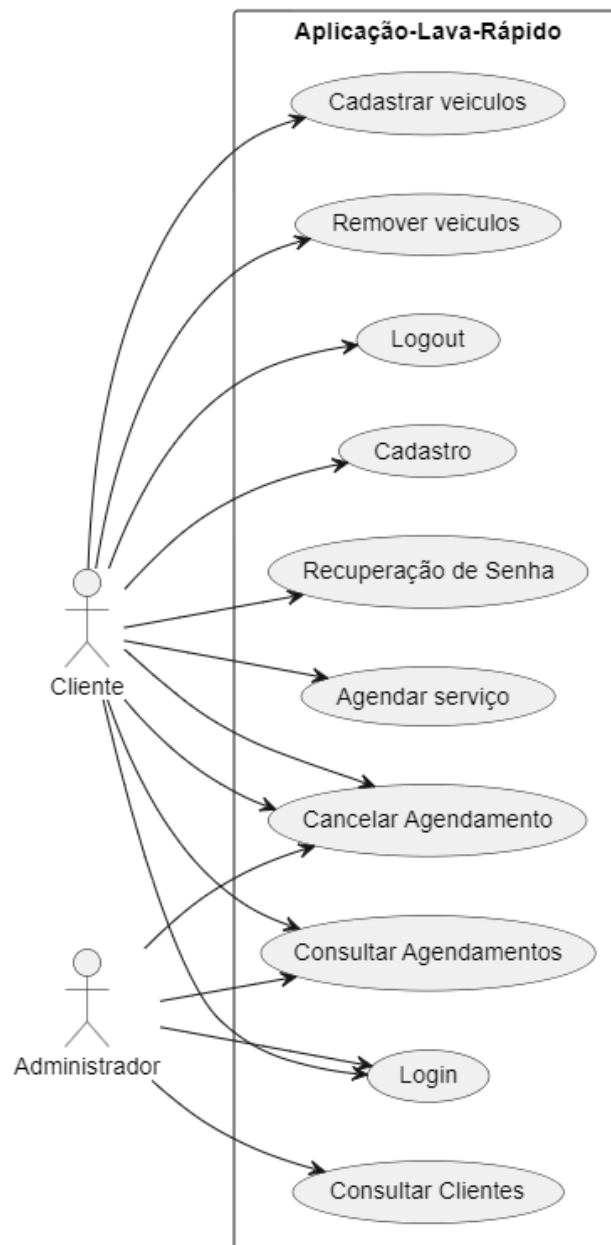
UML (Unified Modeling Language) é uma linguagem padrão usada para especificar, visualizar, construir e documentar os artefatos de sistemas de software. É uma ferramenta amplamente utilizada no desenvolvimento de sistemas orientados a objetos. A UML permite que os desenvolvedores e stakeholders compreendam a estrutura e o comportamento do sistema de forma clara e organizada, utilizando diagramas que representam diferentes aspectos do software.

Os principais tipos de diagramas UML que foram usados no desenvolvimento deste projeto incluem:

3.1.1 Diagrama de Casos de Uso

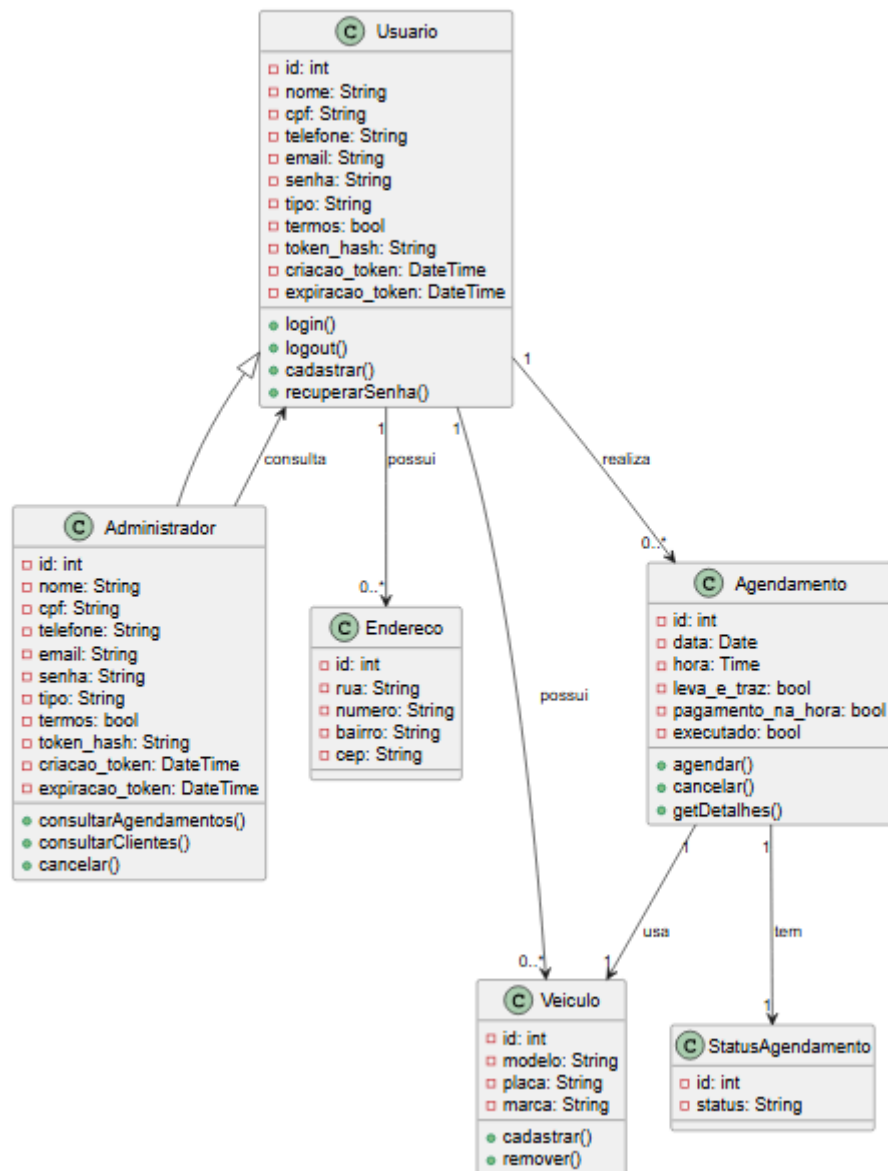
Os casos de uso descrevem as interações típicas dos usuários com o sistema.

Este diagrama mostra os diferentes casos de uso e os atores envolvidos:



3.1.2 Diagrama de Classe

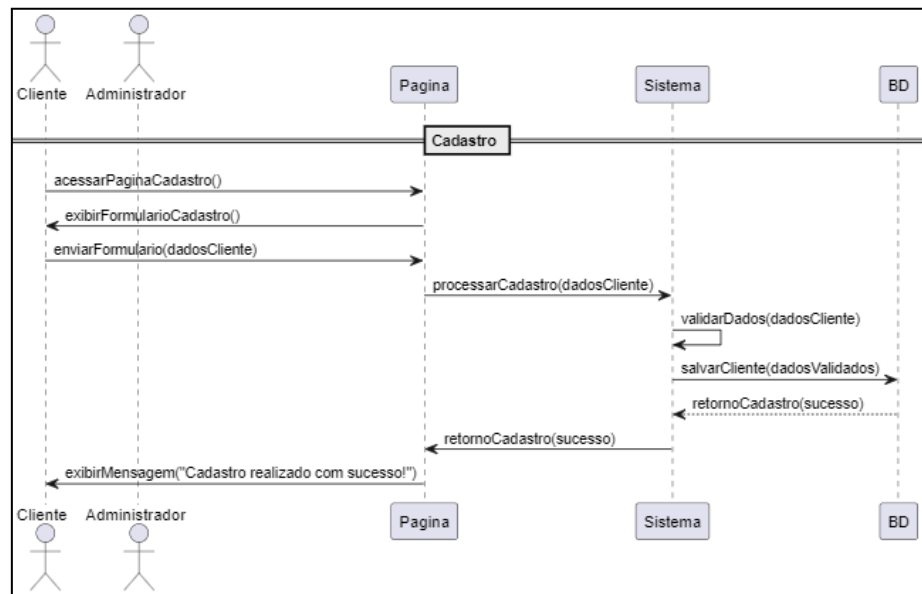
O **Diagrama de Classe** é um tipo de diagrama UML estático que descreve a estrutura de um sistema, mostrando suas classes, seus atributos e os relacionamentos entre elas. Ele oferece uma visão da organização e da arquitetura do software em termos de seus componentes fundamentais e como eles se conectam, a seguir a imagem detalha os componentes do sistema:



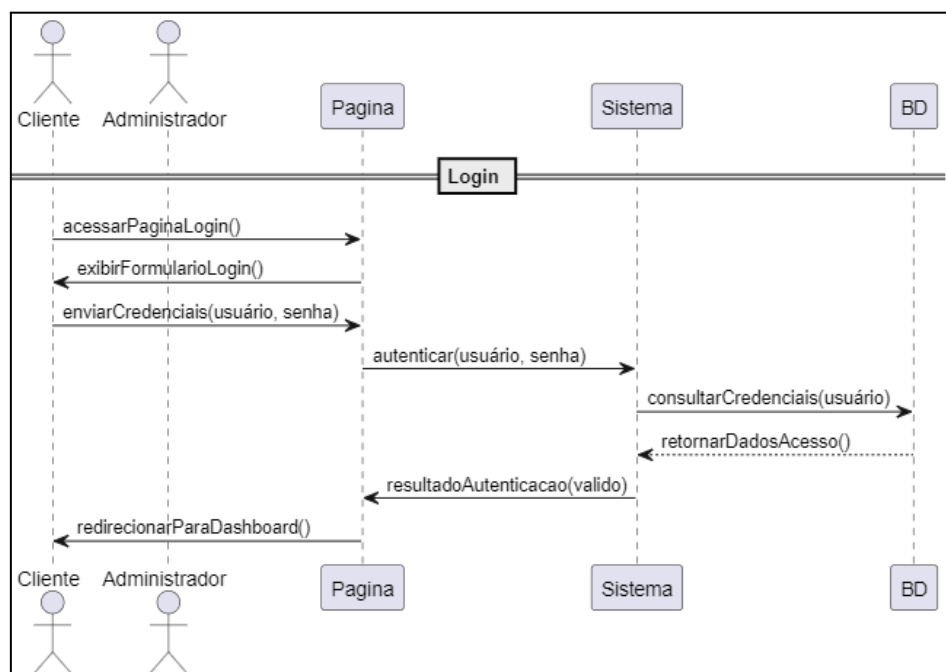
3.1.3 Diagrama de Sequência

O **Diagrama de Sequência** é um tipo de diagrama UML dinâmico que ilustra as interações entre objetos ao longo do tempo para realizar um determinado cenário ou caso de uso. Ele enfatiza a ordem temporal das mensagens trocadas entre os objetos, veja as imagens a seguir:

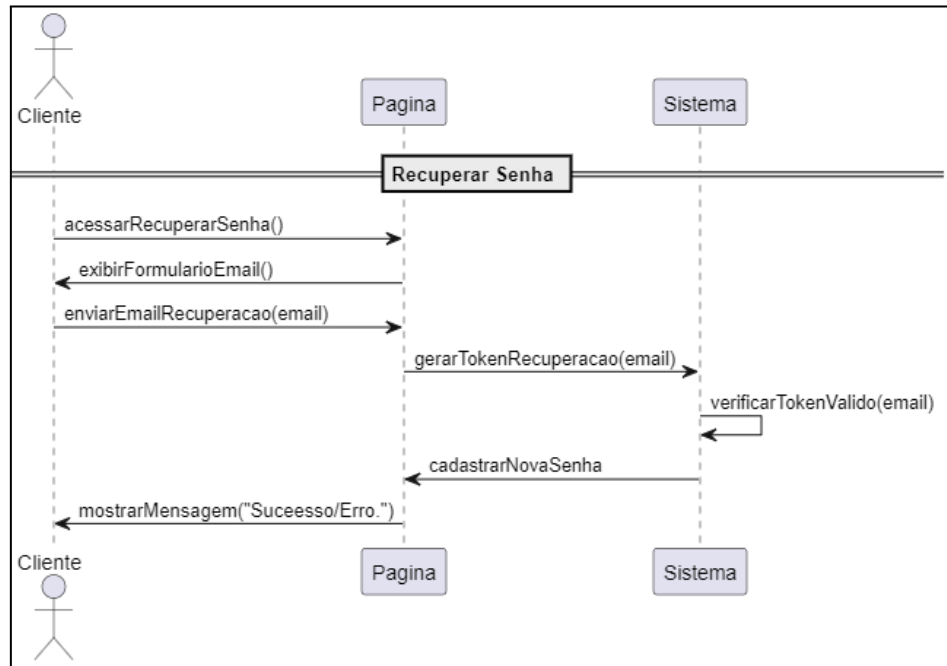
Cadastro:



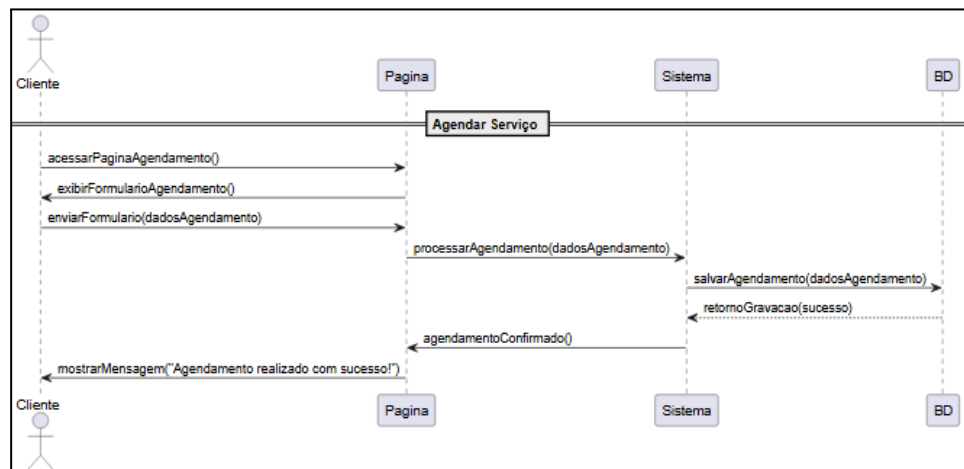
Login:



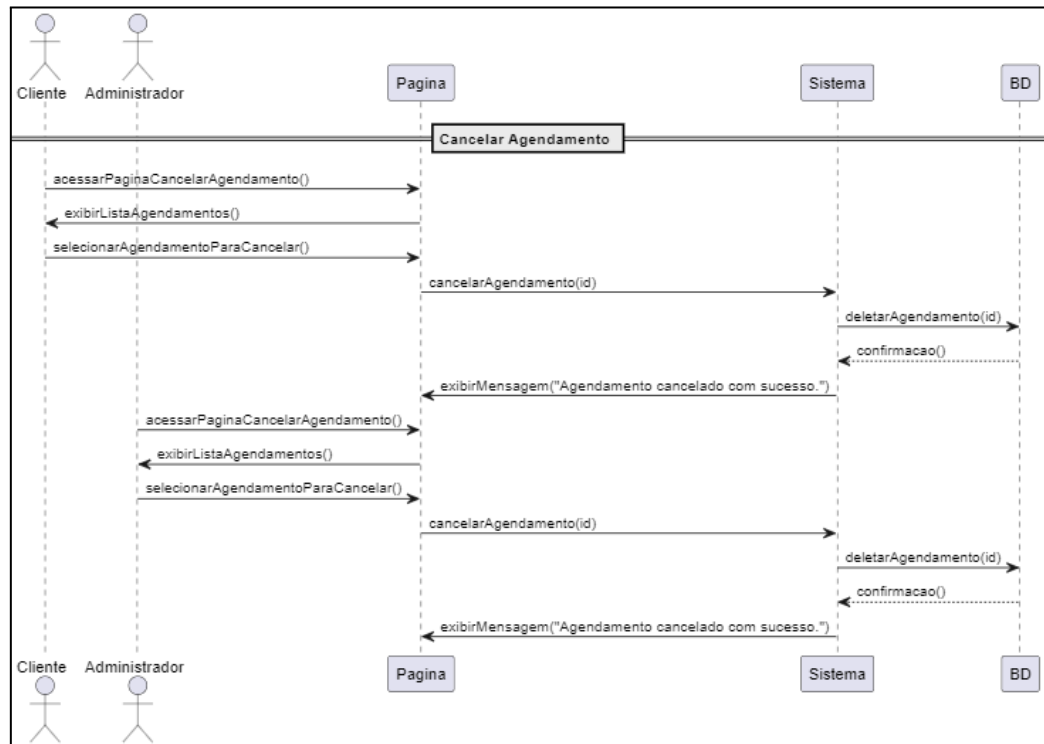
Recuperar Senha:



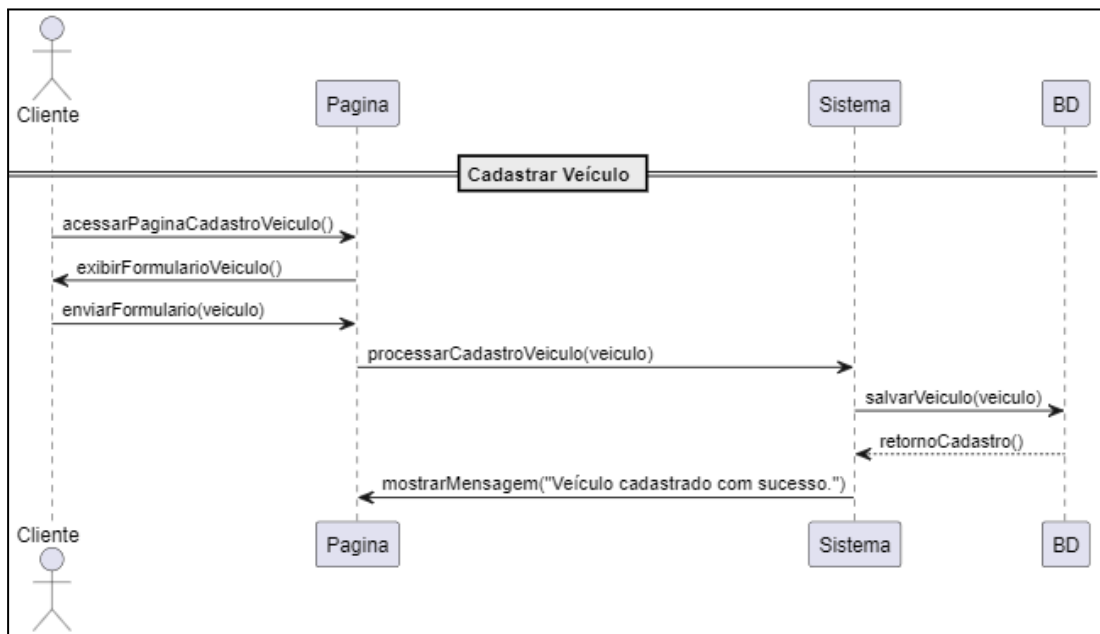
Agendar Serviço:



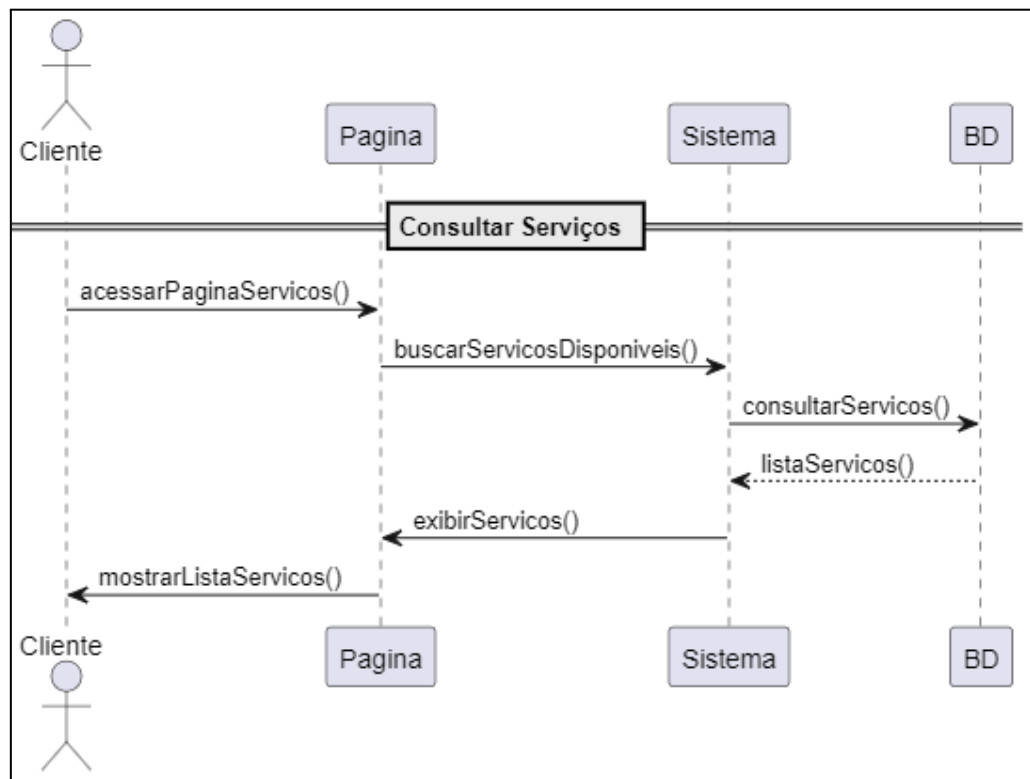
Cancelar Agendamento:



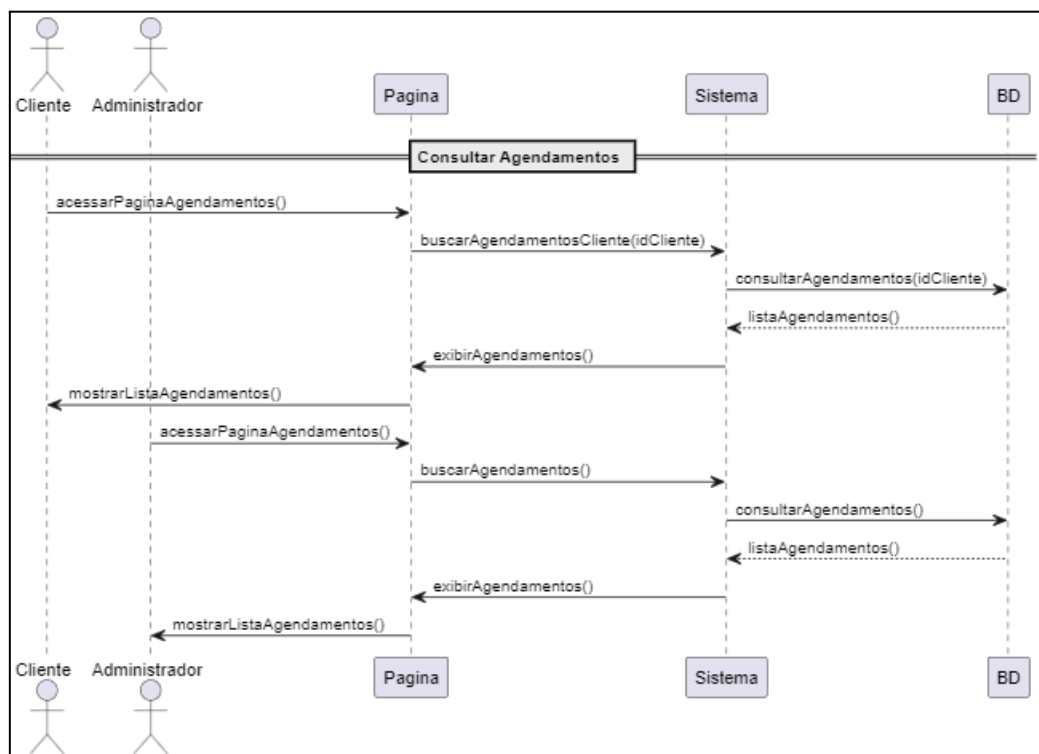
Cadastrar Veículo:



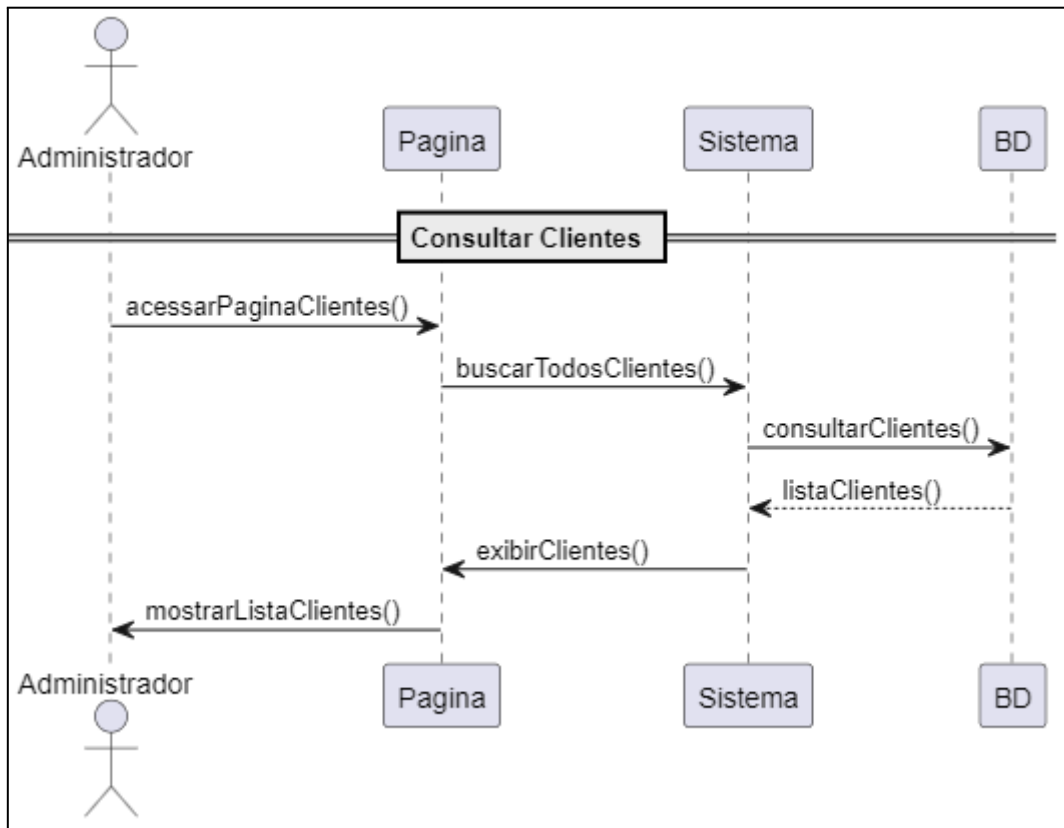
Consultar Serviços:



Consultar Agendamentos:



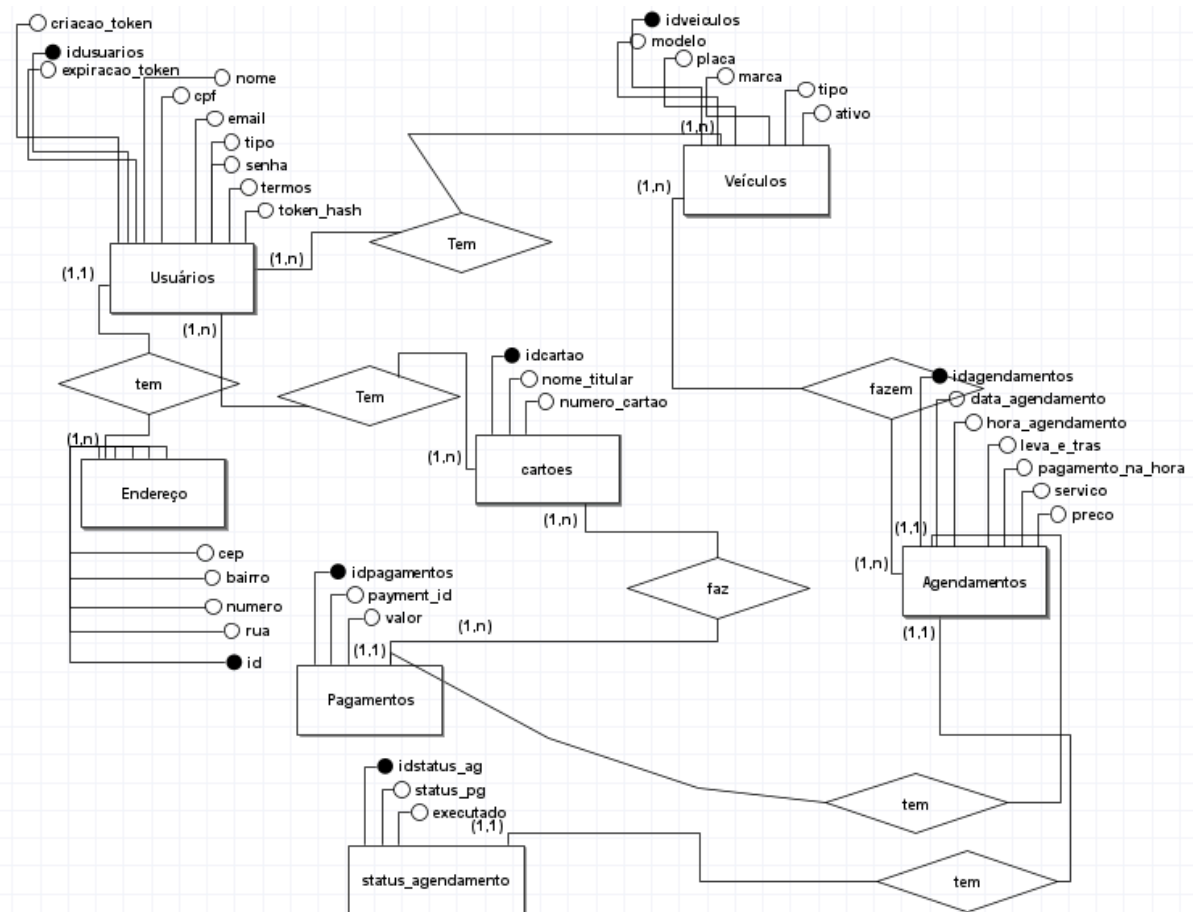
Consultar Clientes (Administrador):



3.2 MODELAGEM DE BANCO DE DADOS

3.2.1 Modelo conceitual

O modelo conceitual de um banco de dados, representando as principais entidades e seus relacionamentos. A Modelagem e criação do banco de dados foi **superdimensionado**, visando implementar os requisitos do plano de evolução do software. O objetivo é fornecer uma compreensão clara da estrutura de dados para o desenvolvimento do sistema, veja a imagem a seguir:



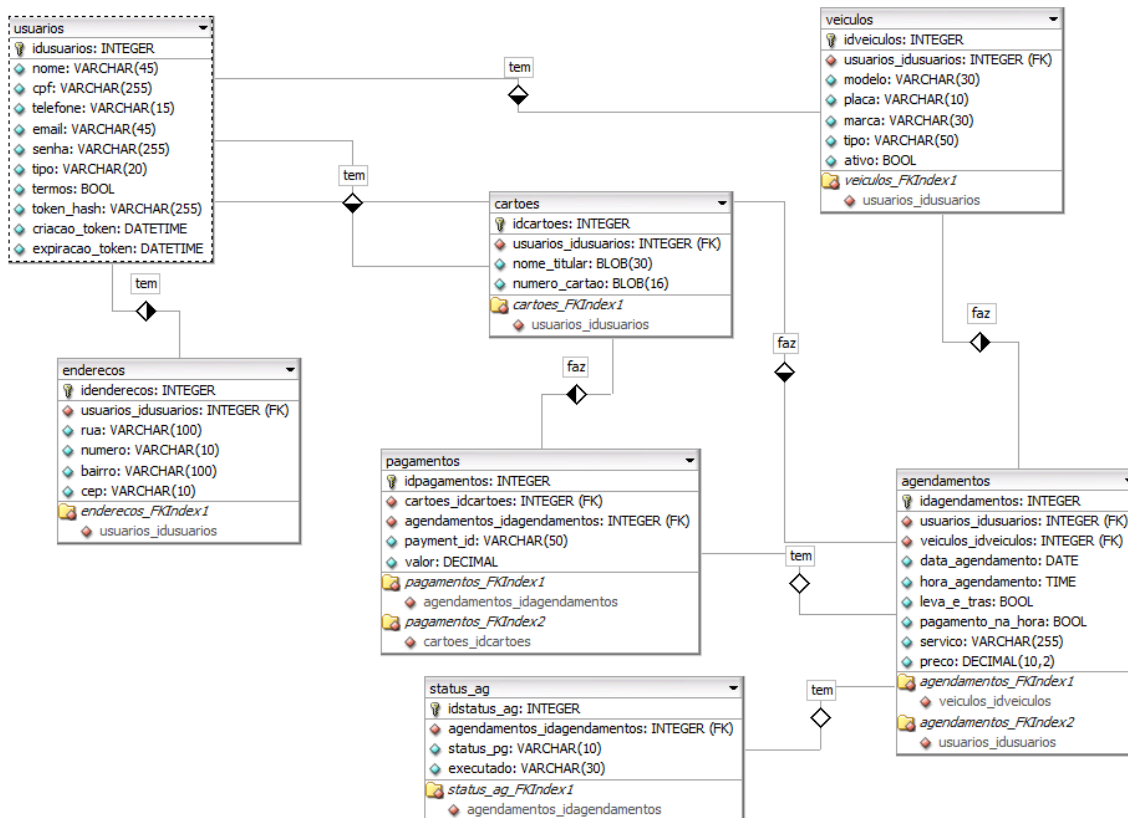
Relacionamentos

Os relacionamentos entre as entidades definem como os dados estão interligados. No modelo apresentado, temos os seguintes relacionamentos:

- **Usuários** possuem **Endereço**: Um usuário pode ter um ou mais endereços (relacionamento 1:n).
- **Usuários** tem **Veículos**: Um usuário pode ter um ou mais veículos (relacionamento 1:n).
- **Usuários** tem **Cartões**: Um usuário pode ter um ou mais cartões (relacionamento 1:n).
- **Veículos** fazem **Agendamentos**: Um veículo pode estar associado a um ou mais agendamentos (relacionamento 1:n).
- **Agendamentos** tem **status_agendamento**: Um agendamento possui um status de agendamento (relacionamento 1:1)..
- **Agendamentos** tem **Pagamentos**: Um agendamento pode estar associado somente a um pagamento (relacionamento 1:1).
- **Cartões** faz **Pagamentos**: Um cartão pode estar associado a um pagamento ou a n pagamentos (relacionamento 1:N).

3.2.2 Modelo lógico

O modelo lógico do banco de dados. Aqui, especificamos os tipos de dados de cada atributo e as chaves estrangeiras que implementam os relacionamentos identificados anteriormente, veja a imagem a seguir:



Chaves Estrangeiras

As chaves estrangeiras estabelecem e reforçam os relacionamentos entre as tabelas:

- A chave estrangeira **usuarios_idusuarios** na tabela **enderecos** referencia a chave primária **idusuarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus endereços.
- A chave estrangeira **usuarios_idusuarios** na tabela **veiculos** referencia a chave primária **idusuarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus veículos.
- As chaves estrangeiras **usuarios_idusuarios** e **veiculos_idveiculos** na tabela **agendamentos** referenciam, respectivamente, as chaves primárias **idusuarios** da tabela **usuarios** e **idveiculos** da tabela **veiculos**, indicando qual usuário agendou um serviço para qual veículo (relacionamento muitos-para-muitos, resolvido pela tabela de agendamentos).

- A chave estrangeira **agendamentos_idagendamentos** na tabela **pagamentos** referencia a chave primária **idagendamentos** da tabela **agendamentos**, indicando a qual agendamento o pagamento está associado (relacionamento um-para-muitos entre agendamentos e pagamentos).
- A chave estrangeira **agendamentos_idagendamentos** na tabela **status_ag** referencia a chave primária **idagendamentos** da tabela **agendamentos**, indicando o status de qual agendamento (relacionamento um-para-um entre agendamentos e status do agendamento).
- A chave estrangeira **usuarios_idusuarios** na tabela **enderecos** referencia a chave primária **idusuarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus endereços.
- A chave estrangeira **usuarios_idusuarios** na tabela **veiculos** referencia a chave primária **idusuarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus veículos.
- As chaves estrangeiras **usuarios_idusuarios** e **veiculos_idveiculos** na tabela **agendamentos** referenciam, respectivamente, as chaves primárias **idusuarios** da tabela **usuarios** e **idveiculos** da tabela **veiculos**, indicando qual usuário agendou um serviço para qual veículo (relacionamento muitos-para-muitos, resolvido pela tabela de agendamentos).
- A chave estrangeira **agendamentos_idagendamentos** na tabela **pagamentos** referencia a chave primária **idagendamentos** da tabela **agendamentos**, indicando a qual agendamento o pagamento está associado (relacionamento um-para-muitos entre agendamentos e pagamentos).
- A chave estrangeira **usuarios_idusuarios** na tabela **cartoes** referencia a chave primária **idusuarios** da tabela **usuarios**, indicando a qual usuário o cartão pertence, (relacionamento um-para-N entre usuários e cartões).

Este detalhamento do modelo lógico fornece as informações necessárias para a implementação física do banco de dados, definindo as tabelas, colunas, tipos de dados e as restrições de integridade referencial.

3.3 RECURSOS

No banco de dados foram implementadas os seguintes objetos no banco de dados, function: `total_agendamentos_hoje()`, para ser utilizada na página: `admin_agendamentos.php`, para retornar o total de agendamentos do dia.

Código que gerou a function no banco de dados:

```

1  -- função de agendamento total do dia.
2  DELIMITER $$
3
4  CREATE FUNCTION total_agendamentos_hoje()
5  RETURNS INT
6  DETERMINISTIC
7  BEGIN
8      DECLARE total INT;
9
10     SELECT COUNT(*) INTO total
11     FROM agendamentos
12     WHERE data_agendamento = CURDATE();
13
14     RETURN total;
15 END$$
16
17 DELIMITER ;
18
19 -- testando a função
20 SELECT total_agendamentos_hoje() AS total_agendamentos;

```

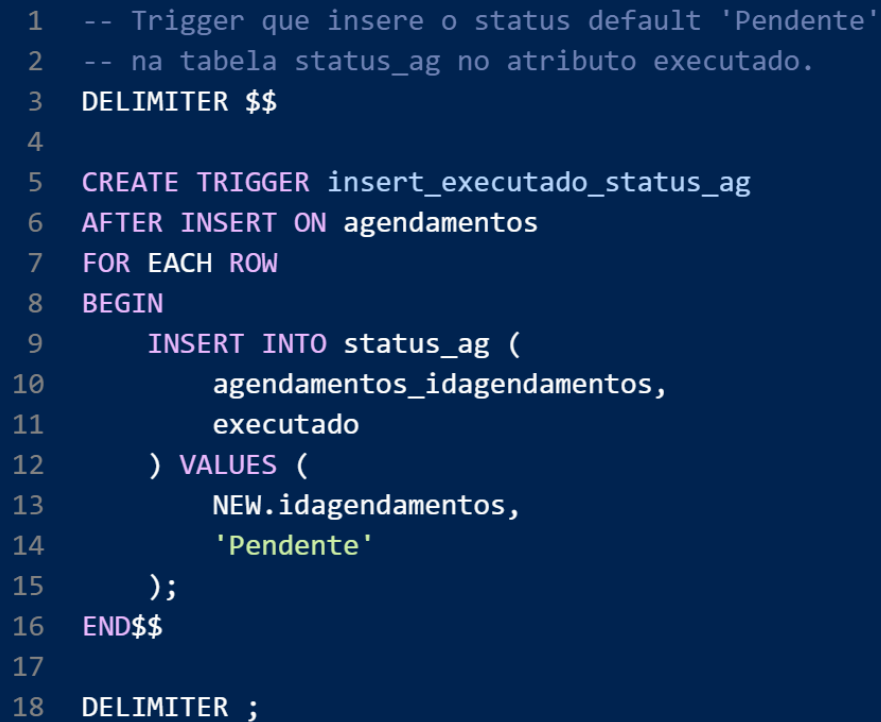
A função criada para a página de agendamentos diários do administrador, é executada no código PHP, da seguinte maneira:

```

1  // Consulta o total de agendamentos do dia via função SQL
2  $sqlTotal = "SELECT total_agendamentos_hoje() AS total_agend";
3  $resultTotal = $conn->query($sqlTotal);
4
5  if ($resultTotal && $rowTotal = $resultTotal->fetch_assoc()) {
6      $total_agend = $rowTotal['total_agend'];
7  } else {
8      $total_agend = 0;
9  }

```

Foi implementado o objeto Trigger: `insert_executado_status_ag`, no banco de dados pois assim que um agendamento é criado a trigger insere o status no atributo executado na tabela `status_ag` como default o dado 'Pendente', veja abaixo o código que criou a Trigger no banco de dados.

A screenshot of a SQL script editor window with a dark blue background and light blue text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The script is a SQL trigger definition. It starts with two comment lines explaining the trigger's purpose. Then it sets the delimiter to '\$\$'. The trigger is named 'insert_executado_status_ag' and is created as an 'AFTER INSERT' trigger on the 'agendamentos' table, firing 'FOR EACH ROW'. The trigger body begins with a 'BEGIN' statement, followed by an 'INSERT INTO' statement into the 'status_ag' table. The columns specified are 'agendamentos_idagendamentos' and 'executado'. The 'VALUES' clause contains 'NEW.idagendamentos' and the string 'Pendente'. The trigger body ends with a semicolon, followed by 'END\$\$'. Finally, the delimiter is reset to a semicolon.

```
1  -- Trigger que insere o status default 'Pendente'
2  -- na tabela status_ag no atributo executado.
3  DELIMITER $$
4
5  CREATE TRIGGER insert_executado_status_ag
6  AFTER INSERT ON agendamentos
7  FOR EACH ROW
8  BEGIN
9      INSERT INTO status_ag (
10         agendamentos_idagendamentos,
11         executado
12     ) VALUES (
13         NEW.idagendamentos,
14         'Pendente'
15     );
16 END$$
17
18 DELIMITER ;
```

O último objeto implementado no banco de dados foi uma procedure para atualizar status de agendamentos para expirado: `update_ag_expirados()`, caso o cliente não apareceu. ou o agendamento ficou para trás sem execução manual do administrador, a procedure insere no atributo executado da tabela `status_ag` referente ao agendamento como expirado, veja o código da procedure implementada:


```

1  -- Procedure para atualizar o status dos agendamentos expirados
2  DELIMITER $$
3
4  CREATE PROCEDURE update_ag_expirados()
5  BEGIN
6      UPDATE status_ag s
7      INNER JOIN agendamentos a ON s.agendamentos_idagendamentos = a.idagendamentos
8      SET s.executado = 'Expirado'
9      WHERE a.data_agendamento < CURDATE()
10     AND s.executado NOT IN ('Concluída', 'Expirado');
11 END $$
12
13 DELIMITER ;

```

Para que a procedure acima seja executada diariamente, criamos o objeto de evento no banco de dados que automatiza a execução da procedure `update_ag_expirados()`, o evento agendado no MySQL executa a procedure, todos os dias às 23:59, automaticamente, veja o código do evento criado no banco mysql:

```

1  -- Evento para tualizar o status dos agendamentos expirados diariamente.
2  DELIMITER $$
3
4  CREATE EVENT IF NOT EXISTS event_update_ag_expired
5  ON SCHEDULE EVERY 1 DAY
6  STARTS CURRENT_DATE + INTERVAL 23 HOUR + INTERVAL 59 MINUTE
7  DO
8  BEGIN
9      CALL update_ag_expirados();
10 END$$
11
12 DELIMITER ;

```

4. DESENVOLVIMENTOS (EXECUÇÃO)

4.1 Banco de dados

O banco de dados utilizado neste projeto será o MySQL. O MySQL é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, conhecido por sua

confiabilidade, desempenho e vasta comunidade de suporte. Ele oferece um conjunto robusto de recursos para o gerenciamento eficiente e seguro dos dados da aplicação.

Versão do MySQL

A versão específica do MySQL a ser utilizada neste projeto é a 10.4.32-MariaDB. MariaDB é um fork da comunidade do MySQL, mantido pela MariaDB Foundation. Esta versão oferece compatibilidade com o MySQL, além de melhorias de desempenho e novos recursos. A escolha desta versão visa garantir a estabilidade e o aproveitamento das funcionalidades mais recentes do SGBD.

Justificativa da Escolha

A escolha do MySQL (versão 10.4.32-MariaDB) para este projeto se baseia nos seguintes fatores:

- **Confiabilidade e Estabilidade:** O MySQL é um SGBD amplamente utilizado e comprovado em diversas aplicações, oferecendo alta confiabilidade e estabilidade para o armazenamento de dados críticos.
- **Desempenho:** O MySQL, especialmente em suas versões mais recentes e forks como o MariaDB, oferece bom desempenho em operações de leitura e escrita, essencial para a responsividade da aplicação.
- **Comunidade e Suporte:** A vasta comunidade de usuários e desenvolvedores do MySQL e MariaDB garante uma ampla disponibilidade de documentação, tutoriais e suporte para solucionar eventuais desafios.
- **Recursos:** O MySQL oferece um conjunto completo de recursos para gerenciamento de dados, incluindo suporte a transações ACID, integridade referencial e diversas opções de indexação.
- **Compatibilidade:** MariaDB é projetado para ser altamente compatível com o MySQL, facilitando a migração e o uso de ferramentas e conhecimentos existentes.

A utilização do MySQL versão 10.4.32-MariaDB proporcionará uma base de dados robusta e eficiente para a aplicação, atendendo aos requisitos de persistência e integridade dos dados.

4.2 Prototipação

Ferramentas Utilizadas

Foi utilizada duas ferramentas muito importantes para a criação dos design do software, **Figma** e **Canvas**.

4.2.1 Descrição dos Protótipos + Interface do usuário

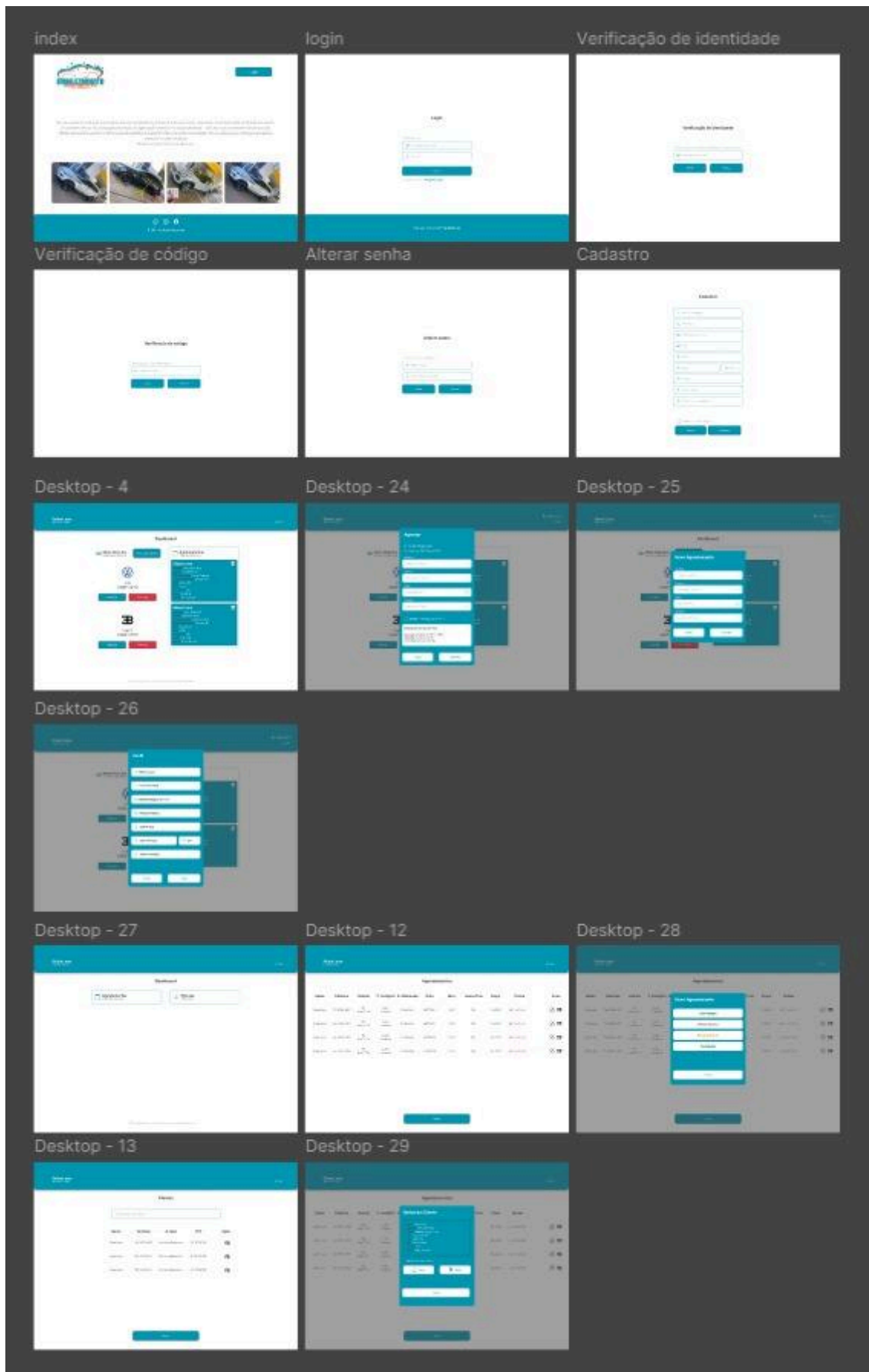
Começamos com uma análise aprofundada do branding para definir o logotipo, paleta de cores e tipografia ideal. Com essa base, avançamos para o design da versão desktop, seguido pela adaptação para mobile. Durante todo o processo, selecionamos imagens de alta qualidade para enriquecer a prototipagem e garantir uma apresentação visual impactante.

4.2.2 Validação dos Protótipo + Feedback

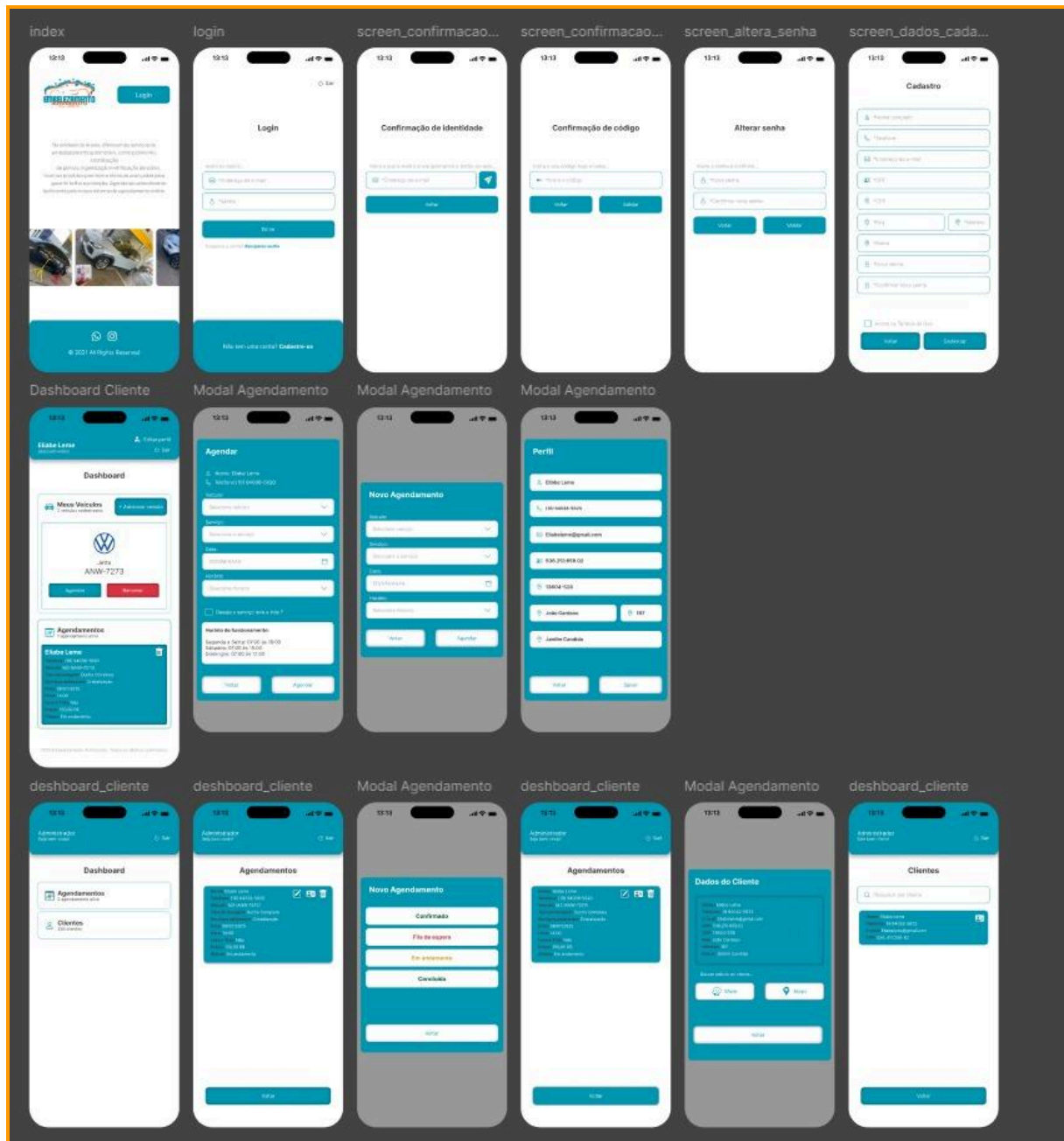
Branding:



Version [desk-top]:



Version [mobile]:



4.3 Codificação

Ambiente de Desenvolvimento

Para o desenvolvimento do software será utilizada a IDE **Visual Studio Code**, em sistema operacional Windows 10 / Windows 11 .

Procedimentos de Desenvolvimento

Os Procedimentos do desenvolvimento do software será em etapas, sendo elas codificação do back end com funcionalidades requisitadas, em um layout temporário enquanto fosse desenvolvido o layout final , na etapa de prototipação; a prototipação estando pronta validada pelo cliente, o novo layout será implementado no projeto já em andamento.

Controle de versionamento (GitHub)

O versionamento da documentação, bem como do código fonte do software, será feito em repositório do [GitHub](#), e será dividido em pastas, uma para documentação e outra para o software-(pasta sistema), após concluído a primeira versão terá um commit com o nome da versão concluída.

Hospedagem

A hospedagem do código fonte, bem como o banco de dados do software da, será feito nos servidores da empresa [hostinger](#), e será custeado pelo cliente.

5. TESTE E QUALIDADE

5.1 Estratégia de testes (descrição de método de teste adotado)

O objetivo da estratégia de teste é garantir que a página da aplicação web do cliente embelezamento automotivo seja apresentada de maneira eficiente, sem vulnerabilidades, e funcional. Isso inclui o navegar entre as páginas somente se o usuário estiver logado no sistema, a estratégia de testes adotada será testes manuais simulando a usabilidade do cliente admin, bem como o cliente convencional, testes em ambiente local com o software [jmeter](#), para simular a quantidades de acessos simultâneos ao software, e testes de vulnerabilidades usando o software [zap](#).

5.2 Resultado dos testes (Apresentação de resultados)

Teste manual

Realizamos testes manuais, para testar todas as rotas da aplicação, e as páginas que necessitam de autenticação do usuário para serem acessadas, além dos testes de todas as funcionalidades da aplicação, o resultado obtido, foi o esperado o usuário para acessar certas rotas da aplicação necessita estar autenticado no sistema, todos os testes de funcionalidades, como cadastro, login/logout, cadastrar veículo/ remover veículo, realizar agendamento/ cancelar agendamento, para cada tipo de usuário, o resultado foi positivo.

No teste feito com o jmeter, para testar quantidade de requisições simultâneas, foram feitas 1000, e outro teste 100 requisições HTTP, do mesmo tipo post.

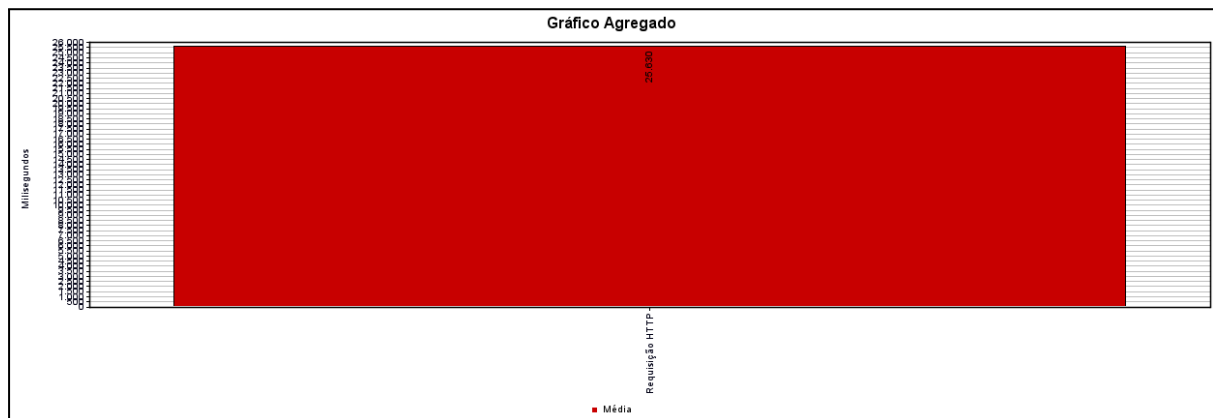
Jmeter Resultados

Resultados com 1000 requisições do tipo POST:

Rótulo	# Amostras	Média	Mediana	90% Line
Requisição HTTP	1000	25630	26802	36224
TOTAL	1000	25630	26802	36224

95% Line	99% Line	Mín.	Máx.	% de Erro
36934	37541	3731	38705	0,00%
36934	37541	3731	38705	0,00%

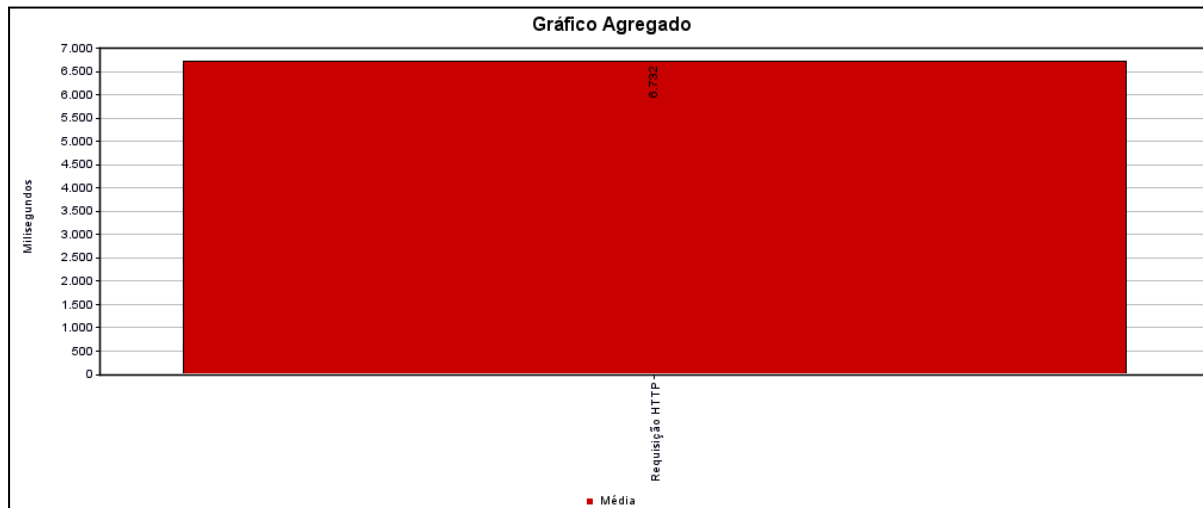
Vazão	KB/s	Sent KB/sec
23,7/sec	205,56	5,00
23,7/sec	205,56	5,00



Resultados com 100 requisições do tipo POST

Rótulo	# Amostras	Média	Mediana	90% Line	95% Line
Requisição ...	100	6732	6655	7561	8078
TOTAL	100	6732	6655	7561	8078

99% Line	Mín.	Máx.	% de Erro	Vazão	KB/s	Sent KB/sec
8207	5573	8239	0,00%	12,0/sec	103,97	2,53
8207	5573	8239	0,00%	12,0/sec	103,97	2,53



O resultado de testes conforme gráficos acima resultou que a última requisição. levou cerca de 25.630 milissegundos em média, no teste de 1000 requisições do tipo POST e o mesmo teste com requisição do tipo GET, levou cerca de 19.268 milissegundos, e no teste POST, com 100 requisições levou cerca de 6.732 milissegundos em média, e o mesmo teste porém com requisição do tipo GET, levou cerca de 3.163 milissegundos a última requisição do teste, considerando sempre a média de tempo da última requisição.

Testes de vulnerabilidades

O Teste feito com o software zap owasp, buscou identificar possíveis vulnerabilidades, veja o resultado do teste abaixo:

zap:

Alertas (12)
> Application Error Disclosure (3)
> Content Security Policy (CSP) Header Not Set (33)
> Missing Anti-clickjacking Header (25)
> Cookie No HttpOnly Flag (3)
> Cookie without SameSite Attribute (3)
> Cross-Domain JavaScript Source File Inclusion (20)
> Divulgação de Data e Hora - Unix (10)
> O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" (24)
> Server Leaks Version Information via "Server" HTTP Response Header Field (81)
> X-Content-Type-Options Header Missing (71)
> Session Management Response Identified (17)
> User Controllable HTML Element Attribute (Potential XSS) (2)

Como bem pode ser observada na imagem acima o resultado de teste de vulnerabilidades **não resultou** em vulnerabilidades de risco alto, como base nas vulnerabilidades encontradas segue a tabela abaixo com a descrição de cada uma delas:

Tabela de Vulnerabilidades - OWASP ZAP

Vulnerabilidade	Risco	Descrição (resumida e traduzida)	Solução (traduzida)
Application Error Disclosure	Médio	A página exibe mensagens de erro com informações sensíveis, como caminho de arquivos.	Revisar o código-fonte. Implementar páginas de erro personalizadas. Registrar erros no servidor com identificador único.
CSP Header Not Set	Médio	Falta da política de segurança de conteúdo que protege contra XSS e injeções.	Configurar o servidor para definir o cabeçalho Content-Security-Policy.
Missing Anti-clickjacking Header	Médio	A página não está protegida contra ataques de clickjacking.	Incluir cabeçalhos Content-Security-Policy: frame-ancestors 'none'; ou X-Frame-Options.
Cookie No HttpOnly Flag	Baixo	Cookie acessível via JavaScript, vulnerável a roubo por scripts maliciosos.	Definir o atributo HttpOnly para todos os cookies.
Cookie without SameSite Attribute	Baixo	Cookies vulneráveis a requisições entre sites (CSRF).	Definir o atributo SameSite como Lax ou Strict.
Cross-Domain JavaScript Source File Inclusion	Baixo	Scripts carregados de domínios externos.	Garantir que apenas fontes confiáveis sejam usadas para scripts.
Divulgação de Data e Hora - Unix	Baixo	A aplicação expõe carimbos de data/hora.	Confirmar se esses dados não são sensíveis ou exploráveis.
Cabeçalho "X-Powered-By" exposto	Baixo	Cabeçalho revela tecnologia usada (ex: PHP, ASP.NET).	Configurar servidor para ocultar cabeçalho X-Powered-By.
Cabeçalho "Server" expõe versão	Baixo	Cabeçalho Server revela versão do servidor web.	Ocultar ou tornar genérico o cabeçalho Server.
X-Content-Type-Options Header Missing	Baixo	Cabeçalho que impede sniffing de MIME ausente.	Adicionar cabeçalho X-Content-Type-Options: nosniff.
Session Management Response Identified	Informacional	Resposta contém token de sessão (ex: PHPSESSID).	Alerta informativo, sem ação necessária.

Como bem pode ser observado na tabela acima as principais vulnerabilidades encontradas foram relacionadas a configuração de servidor, o teste foi realizado em ambiente de desenvolvimento, e quando for colocado em produção terá que ser aplicada às prevenções de vulnerabilidades relacionadas a configuração de servidor, as vulnerabilidades encontradas referentes a isso foram:

- CSP Header Not Set
- Cabeçalho "X-Powered-By" exposto
- Cabeçalho "Server" expõe versão
- Missing Anti-clickjacking Header
- X-Content-Type-Options Header Missing

Foram encontradas vulnerabilidades relacionadas a Cookies, apesar do sistema não utilizar Cookies para armazenamento de informações no navegador, porém provavelmente algum cookie está sendo gerado automaticamente por, sessão do PHP (PHPSESSID), alguma biblioteca de terceiros, ou pelo servidor (Apache/Nginx, proxy reverso, load balancer), as vulnerabilidades relacionadas a isso foram:

- Cookie No HttpOnly Flag
- Cookie without SameSite Attribute

Solução para a vulnerabilidade, foi configurar parâmetros da sessão no arquivo init.php (logo antes de session_start()), em conjunto com geração, inserção com formulários, validação nos scripts PHP que recebem requisição do tipo POST, e autenticação de token do tipo CSRF para prevenir contra a vulnerabilidade:

```

1 // Inicia a sessão:
2 if (session_status() === PHP_SESSION_NONE) {
3     // Configurações de segurança para a sessão.
4     ini_set('session.cookie_httponly', 1); // Impede acesso ao cookie via JS
5     ini_set('session.cookie_secure', 1); // Garante que só trafegue via HTTPS
6     ini_set('session.cookie_samesite', 'Strict'); // Protege contra ataques cross-site (CSRF)
7     session_start(); // Inicia a sessão se ainda não estiver iniciada.
8
9 }
```

Foi encontrada vulnerabilidade considerada de risco baixo relacionada com scripts externos, a vulnerabilidade encontrada foi a:

- Cross-Domain JavaScript Source File Inclusion

Quanto à solução da vulnerabilidade de risco baixo encontrada, não há muito a ser feito pois o sistema utiliza CDN's tanto do bootstrap 5 e da CDN da biblioteca Sweet Alert.

5.3 Garantia da qualidade (práticas adotadas)

Com o objetivo de assegurar a qualidade e a segurança da aplicação web desenvolvida, foram adotadas práticas voltadas especialmente à revisão de código e à correção de vulnerabilidades identificadas ao longo do projeto.

- Revisão contínua do código-fonte, com foco na validação de entradas do usuário, prevenção de injeção de código malicioso e uso de práticas seguras de desenvolvimento, como criptografia de dados sensíveis (ex.: senhas e CPF).

- Testes automatizados com JMeter, para simular múltiplos acessos simultâneos e verificar a estabilidade da aplicação sob carga, garantindo desempenho aceitável mesmo em picos de acesso.

- Revisões cruzadas entre os membros da equipe, onde cada integrante testava funcionalidades desenvolvidas por outro, com foco tanto na usabilidade quanto na identificação de falhas de segurança.

Essas práticas contribuíram para fortalecer a segurança, confiabilidade e qualidade geral da aplicação, garantindo uma base sólida para sua evolução futura.

5.4 Requisitos Mínimos de Hardware e Software

5.4.1. Requisitos do Usuário Final

Estes requisitos referem-se aos dispositivos utilizados pelos usuários (clientes e administradores) para acessar o sistema via navegador.

Hardware

- Processador: Mínimo Dual-core 1.6 GHz
- Memória RAM: Mínimo 2 GB (4 GB recomendado para melhor desempenho)
- Armazenamento disponível: Mínimo 100 MB livres (para cache e dados temporários do navegador)
- Resolução da tela: Mínimo 1280 x 720 pixels
- Conectividade: Conexão estável com velocidade mínima de 2 Mbps

Software

- Sistema Operacional compatível:
 - Windows 10 ou superior
 - Android 9.0 ou superior

- iOS 13 ou superior
 - Distribuições Linux com suporte a navegadores modernos
- Navegadores compatíveis:
 - Google Chrome versão 90 ou superior
 - Mozilla Firefox versão 88 ou superior
 - Microsoft Edge versão 90 ou superior
 - Safari versão 13 ou superior
- Outros requisitos:
 - JavaScript habilitado
 - Navegador com suporte a HTML5 e CSS3

5.4.2. Requisitos do Servidor (Hospedagem)

Estes requisitos referem-se ao ambiente onde o sistema será hospedado e executado.

Hardware

- Processador: Mínimo 1 vCPU (recomendado 2 vCPUs ou mais)
- Memória RAM: Mínimo 1 GB (recomendado 2 GB ou mais)
- Armazenamento em disco: Mínimo 10 GB livres
- Conexão de rede: Largura de banda estável com no mínimo 100 Mbps

Software

- Sistema Operacional: Linux (Ubuntu, Debian, CentOS ou equivalente)
- Servidor Web: Apache 2.4 ou superior, ou Nginx

- Versão do PHP: PHP 7.4 ou superior (recomendada PHP 8.x)
- Dependências PHP necessárias:
 - `mysqli`
 - `PHP Mailer`
 - `Dotenv`
 - `openssl`
- Banco de Dados: MySQL 5.7 ou superior, ou MariaDB 10.x
- Gerenciador de dependências: Composer (se aplicável)
- Segurança: Certificado SSL (HTTPS) recomendado para comunicação segura

6. CONTRATO PARA DESENVOLVIMENTO DO SOFTWARE

É de extrema importância ressaltar que o contrato foi formulado e celebrado com o cliente, para evitar problemas futuros, já para o sistema em produção; o contrato se encontra junto à documentação, e denominado de Anexo III.

7. PLANO DE EVOLUÇÃO

O plano de evolução inclui futuras melhorias e funcionalidades adicionais, como:

ID	Funcionalidade Planejada	Descrição	Prioridade	Observações Técnicas
EV01	Desconto para horários próximos	Aplicar desconto automático em horários com pouca demanda.	Média	Requer lógica de tempo e regra de precificação.
EV02	Confirmação de endereço de busca	Perguntar ao cliente se o endereço de busca do veículo é o mesmo cadastrado. Permitir alteração temporária e reverter após agendamento.	Alta	Necessita salvar endereço temporário e restaurar após status "Concluído".

EV03	Limitar cadastro de veículos	Restringir quantidade de veículos por cliente para evitar excesso de registros.	Alta	Verificar regra de limite ideal (ex: 5 veículos).
EV04	Gráficos no Dashboard do administrador	Criar painel com gráficos de agendamentos, serviços realizados, faturamento, etc.	Alta	Pode usar Chart.js, Recharts ou Google Charts.
EV05	Notificação de busca do carro	Avisar o administrador 15 minutos antes do horário de agendamento para buscar o carro.	Alta	Necessita cron job ou serviço agendado.
EV06	Visualização por calendário (Admin)	Substituir visualização atual por uma agenda interativa em formato de calendário semanal.	Alta	Sugestão de uso: FullCalendar.js
EV07	Integração com API do Detran	Preenchimento automático dos dados do veículo via placa.	Média	Depende da disponibilidade da API do Detran do estado.
EV08	Configuração do PagBank	Adicionar tela de configuração de credenciais da API do PagBank no painel administrativo.	Alta	Guardar dados criptografados.
EV09	Cadastro de tabela de serviços	Criar tabela no banco para serviços e permitir edição futura dos dados.	Alta	Relacionar com agendamento e valores.
EV10	Imagem de perfil e notificações	Permitir upload de imagem de perfil e configurar notificações futuras.	Média	Exigir redimensionamento e compressão da imagem.

EV11	Pesquisa de clientes (Admin)	Adicionar campo de busca por nome, e-mail ou telefone no painel do administrador.	Alta	Implementar busca com filtro incremental (live search).
EV12	Notificações para o usuário	Enviar lembretes de agendamento, confirmações e status por push ou e-mail.	Alta	Pode usar serviços como Firebase ou notificações por e-mail.
EV13	Edição de valores de serviços (Admin)	Criar tela para o administrador editar valores dos serviços disponíveis.	Alta	Interface simples ligada à tabela de serviços (EV09).
EV14	Pagamento via PagBank	Implementar gateway de pagamento via PagBank.	Alta	Requer criação de pedidos, retorno de status e conciliação.
EV15	Sessão “Trabalhe Conosco” no site	Adicionar uma seção no website para que candidatos possam enviar currículos ou preencher formulário de interesse.	Média	Necessário criar nova rota pública e formulário com e-mail ou banco.
EV16	Envio aviso via whatsapp	Enviar um lembrete para o usuário com antecedência de 30 minutos do agendamento	Baixa	Requer criação de lógica e verificar bibliotecas disponíveis.

A versão inicial visou cumprir o objetivo que era disponibilizar uma aplicação web funcional para a empresa Embelezamento Automotivo, com foco na automatização do processo de agendamento de serviços de lavagem automotiva. Essa primeira entrega contemplou as funcionalidades essenciais, como cadastro e login de clientes, visualização dos serviços oferecidos, agendamento com escolha de data e horário, além de um painel administrativo para gerenciamento dos agendamentos e clientes.

A aplicação foi desenvolvida com uma interface responsiva, adaptada para diferentes dispositivos, garantindo acessibilidade e facilidade de uso tanto para os clientes quanto para os administradores. Todas as decisões de escopo foram guiadas pela viabilidade técnica, tempo disponível e objetivos pedagógicos do projeto.

Com a entrega desta versão inicial, o sistema alcança sua proposta fundamental de organizar e facilitar os agendamentos. A partir disso, o projeto está apto a evoluir com novas funcionalidades e melhorias, de acordo com as demandas futuras do cliente, como relatórios mais detalhados, notificações automatizadas, integração com outras ferramentas e aprimoramentos na gestão administrativa.

8. REFERÊNCIAS

Anexo I - Diário de bordo- físico

Anexo II - Cronograma efetivo

Anexo III – Contrato

- **PHP Manual – Requisitos de Sistema**

PHP Group. (2024). *PHP Manual: Requisitos de instalação*. Disponível em: https://www.php.net/manual/pt_BR/install.general.php (Usada para as versões mínimas recomendadas de PHP e extensões necessárias)

- **Apache HTTP Server Documentation**

The Apache Software Foundation. (2024). *Apache HTTP Server Version 2.4 Documentation*. Disponível em: <https://httpd.apache.org/docs/> (Referência para o uso do Apache como servidor web)

- **Nginx Documentation**

NGINX, Inc. (2024). *Nginx Documentation*. Disponível em: <https://docs.nginx.com/> (Referência para uso do servidor web Nginx)

- **MySQL – Requisitos de Hardware e Software**

Oracle Corporation. (2024). *MySQL 5.7 Reference Manual*. Disponível em: <https://dev.mysql.com/doc/refman/5.7/en/> (Referência para requisitos e instalação do MySQL)

- **Composer – Dependency Management for PHP**

Composer Team. (2024). *Composer Documentation*. Disponível em: <https://getcomposer.org/doc/> (Referência para o uso do Composer como gerenciador de dependências)

- **DigitalOcean – Requisitos para Hospedagem Web**

DigitalOcean. (2024). *Droplets Requirements and Configuration*. Disponível em: <https://docs.digitalocean.com/products/droplets/> (Usado como base para definir CPU, RAM, e largura de banda mínima para servidores de aplicações PHP)

- **Google Chrome – Requisitos de Sistema**

Google LLC. (2024). *Requisitos do sistema para o navegador Chrome*. Disponível

em: <https://support.google.com/chrome/a/answer/7100626?hl=pt-BR> (Referência para compatibilidade mínima do navegador)

- **Mozilla Firefox – Requisitos de Sistema**

Mozilla Foundation. (2024). *Requisitos do sistema do Firefox*. Disponível em: <https://www.mozilla.org/pt-BR/firefox/system-requirements/> (Referência para versões suportadas do Firefox)

- **W3C – HTML5 e CSS3 Standards**

World Wide Web Consortium (W3C). (2024). *HTML5 and CSS Specifications*. Disponível em: <https://www.w3.org/standards/> (Referência para os padrões utilizados por navegadores modernos)

- **LGPD – Lei Geral de Proteção de Dados Pessoais (Brasil)**

Presidência da República. (2018). *Lei nº 13.709, de 14 de agosto de 2018*. Disponível em: <https://www.gov.br/cidadania/pt-br/aceso-a-informacao/lgpd> (Reforça os aspectos de segurança e privacidade dos dados dos usuários)