

**Fatec Araras “Antônio Brambilla”**

***Documentação de Software***

***Lava Rápido***

***Projeto Integrador - 2º Semestre - DSM***

| ***Integrantes*** |
| --- |
| *Lucas Eduardo Rosolem* |
| *Bruno Eduardo Rodrigues* |
| *Renan Augusto Eugenio Marques* |
| *Willian Adriano Dias Paulino* |
|  |
|  |

## **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 |  |
| 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 | ? |
|  | Deploy FrontEnd | ? | ? |

## 

## Sumário

## **0.0 Cronograma Geral do Projeto**

## **INTRODUÇÃO** 1.1 Apresentação da Empresa 1.2 Missão, Visão e Valores 1.3 Repositório

## **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)

## **ESPECIFICAÇÃO DE SOFTWARE** 3.1 Metodologia de Desenvolvimento 3.2 Diagramas UML 3.2.1 Diagrama de Caso de Uso 3.2.2 Diagrama de Classe 3.2.3 Diagrama de Sequência 3.3 Modelagem de Banco de Dados 3.3.1 Modelo Conceitual 3.3.2 Modelo Lógico 3.4 Recursos

## **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

## **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 para usabilidade do sistema

## **CONTRATO PARA DESENVOLVIMENTO DE SOFTWARE**

## **PLANO DE EVOLUÇÃO**

## **REFERÊNCIAS** Anexo I - Diário de bordo Anexo II – Cronograma efetivo

**1. INTRODUÇÃO**

***1.1 Apresentação da empresa***

Este documento descreve a especificação do projeto de uma aplicação web, para a gestão de agendamentos e pagamentos de lavagens 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 e pagamento dos serviços oferecidos.

A aplicação será implementada para atender às necessidades do cliente real **Embelezamento Automotivo**, um lava-jato 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 e efetuem pagamentos de forma prática e segura, 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-semestre: Sistema de Lava Rápido**](https://github.com/Lucas-Ed/Pi-2-semestre), o projeto, apesar de ser público detêm 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 e pagamento de serviços de lavagem automotiva para o cliente **Embelezamento Automotivo**. A solução está sendo desenvolvida pelo grupo **Web Lab**, como parte do Projeto Integrador do curso de **Desenvolvimento de Sistemas Multiplataformas** da **Fatec Araras**.

A aplicação terá como foco principal a automatização do processo de agendamento e a integração com um meio de pagamento online, 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 lavagens 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 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** e **pagamento online** 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 efetuar pagamentos de forma rápida e segura 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-jato) 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*** | |
| --- | --- |
| ***Definição*** | ***Login*** |
| ***Ator*** | ***Administrador*** |
| ***Prioridade*** | ***Essencial.*** |
| ***Requisitos Não Funcionais Associados*** | ***RNF - 001 / RNF - 003*** |
| ***Código*** | ***Descrição*** |
| ***RF - 001*** | ***O sistema deve permitir que 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*** | ***Cancelar Agendamento*** |
| ***Ator*** | ***Administrador*** |
| ***Prioridade*** | ***Essencial.*** |
| ***Requisitos Não Funcionais Associados*** | ***RNF - 004 / RNF - 005*** |
| ***Código*** | ***Descrição*** |
| ***RF - 002*** | ***O sistema deve permitir que o administrador cancele um agendamento existente.*** |
|  |  |
| ***Definição*** | ***Consultar Agendamentos*** |
| ***Ator*** | ***Administrador*** |
| ***Prioridade*** | ***Essencial.*** |
| ***Requisitos Não Funcionais Associados*** | ***RNF - 006 / RNF - 007*** |
| ***Código*** | ***Descrição*** |
| ***RF - 003*** | ***O sistema deve permitir que o administrador consulte a lista de agendamentos, com possibilidade de filtragem por data, cliente ou status.*** |
|  |  |
| ***Definição*** | ***Consultar Clientes*** |
| ***Ator*** | ***Administrador*** |
| ***Prioridade*** | ***Essencial.*** |
| ***Requisitos Não Funcionais Associados*** | ***RNF - 008 / RNF - 009*** |
| ***Código*** | ***Descrição*** |
| ***RF - 004*** | ***O sistema deve permitir que o administrador consulte informações dos clientes cadastrados.*** |

***2.4 Requisitos não funcionais (RNF)***

| **Requisitos Não Funcionais** | |
| --- | --- |
| **Código** | **Descrição** |
| **RNF - 001** | **O tempo de resposta do login não deve ultrapassar 3 segundos.** |
| **RNF - 002** | **As senhas e CPF’s devem ser armazenadas de forma criptografada.** |
| **RNF - 003** | **A sessão deve expirar após 15 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.** |

**3 ESPECIFICAÇÃO DE SOFTWARE**

***3.1 Metodologia de desenvolvimento***

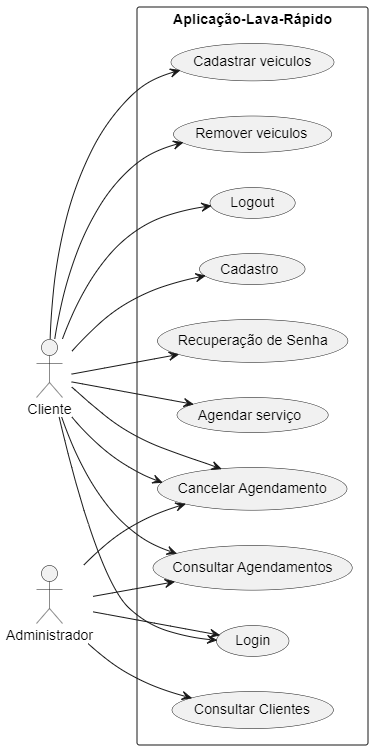
***3.2 .Diagramas UML*** UML (Unified Modeling Language) é uma linguagem padrão usada para especificar, visualizar, construir e documentar os artefatos de sistemas de software. É uma ferramenta amplamente utilizada no desenvolvimento de sistemas orientados a objetos. A UML permite que os desenvolvedores 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:

***3.2.1 Diagrama de Casos de Uso***

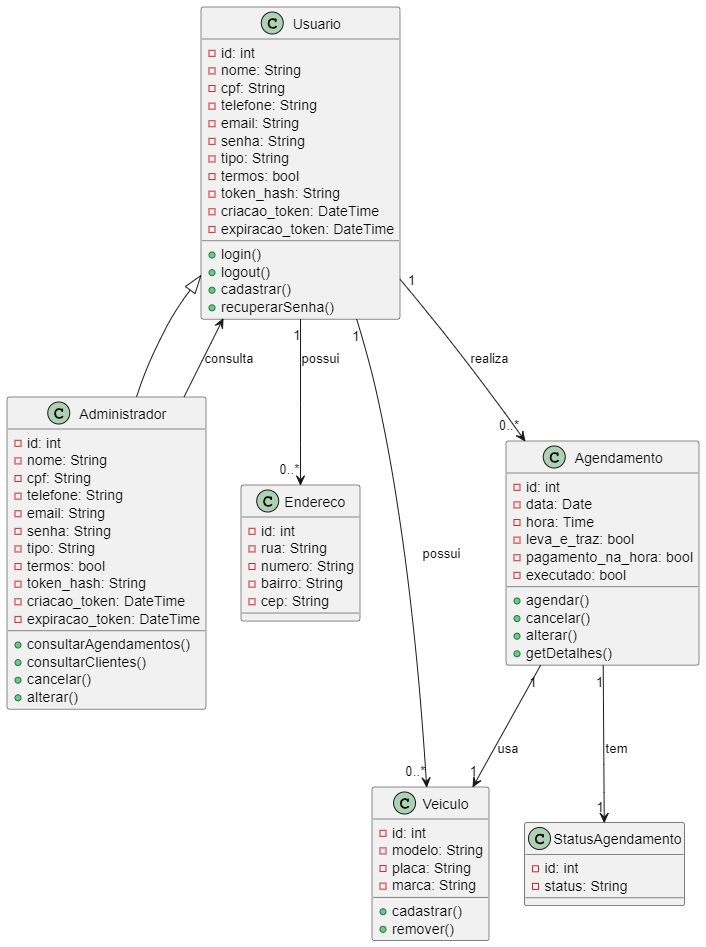
Os casos de uso descrevem as interações típicas dos usuários com o sistema.

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



***3.2.2 Diagrama de Classe***

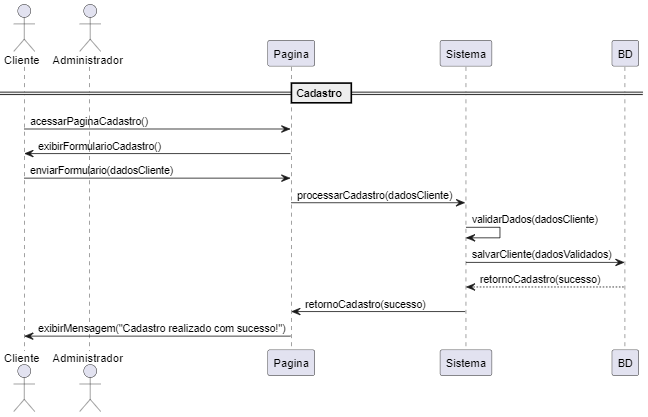
O **Diagrama de Classe** é um tipo de diagrama UML estático que descreve a estrutura de um sistema, mostrando suas classes, seus atributos e os relacionamentos entre elas. Ele oferece uma visão da organização e da arquitetura do software em termos de seus componentes fundamentais e como eles se conectam,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



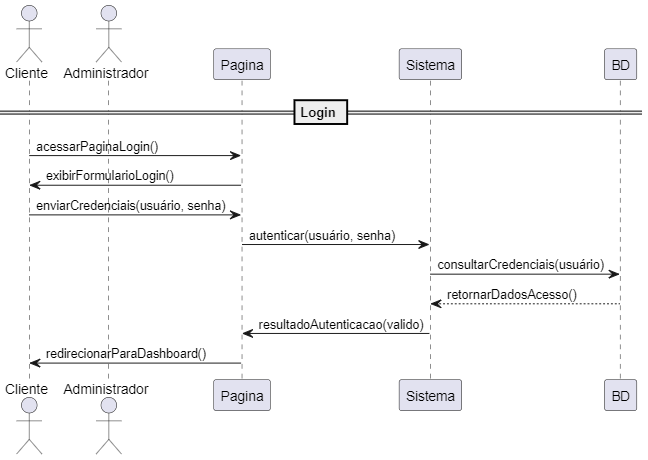
***3.2.3 Diagrama de Sequência***

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:

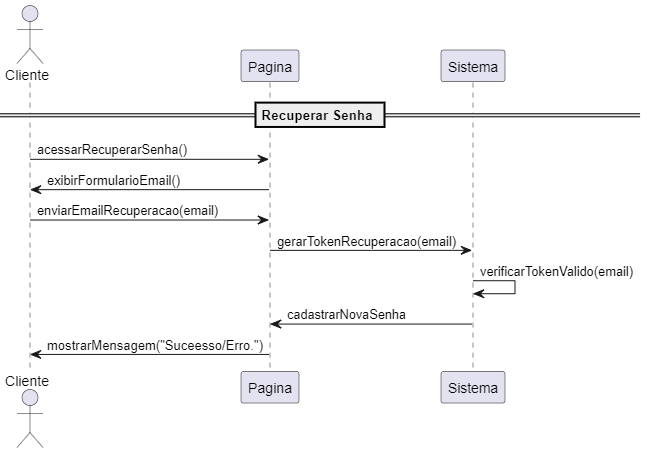
**Cadastro:**

****

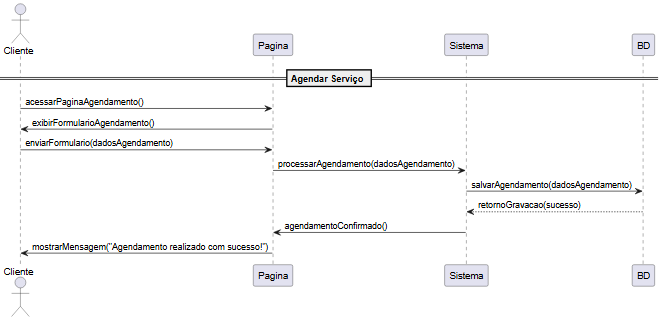
**Login:**

****

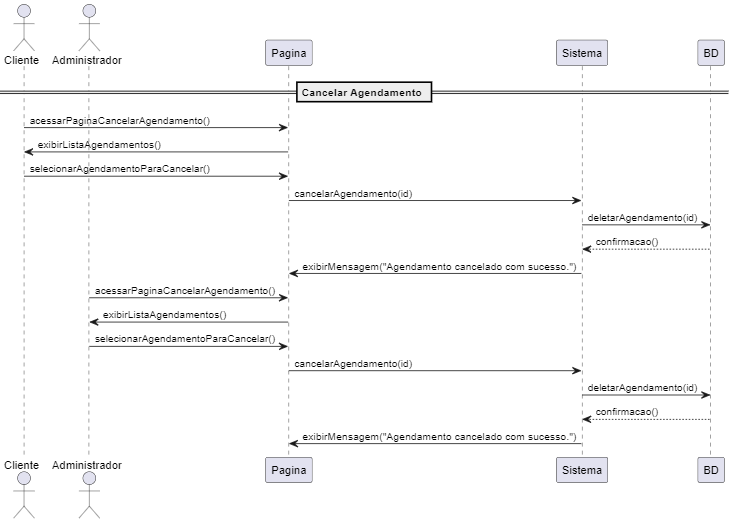
**Recuperar Senha:**

****

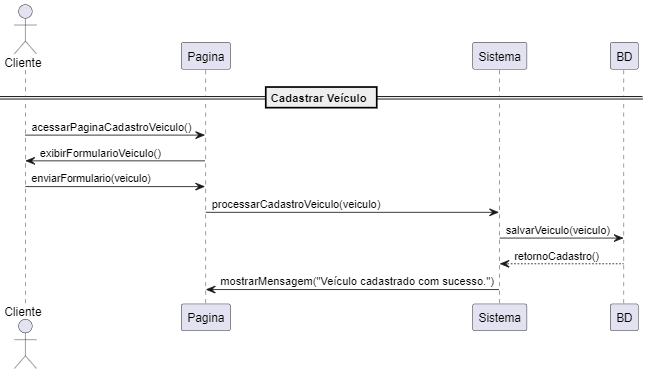
**Agendar Serviço:**

****

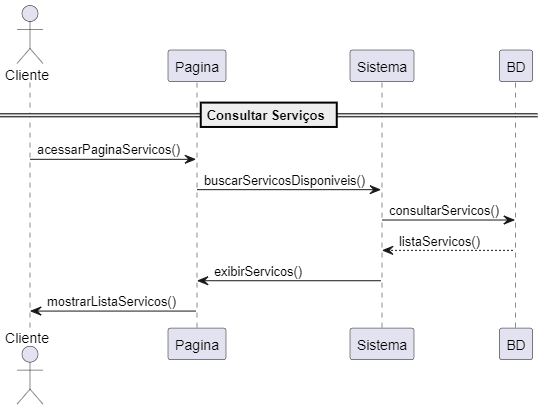
**Cancelar Agendamento:**

****

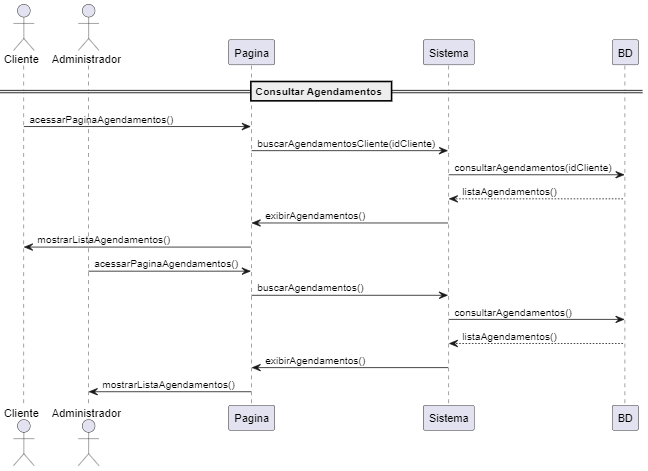
**Cadastrar Veículo:**

****

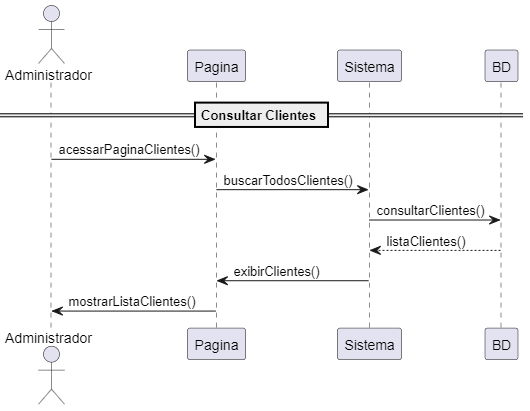
**Consultar Serviços:**

****

**Consultar Agendamentos:**

****

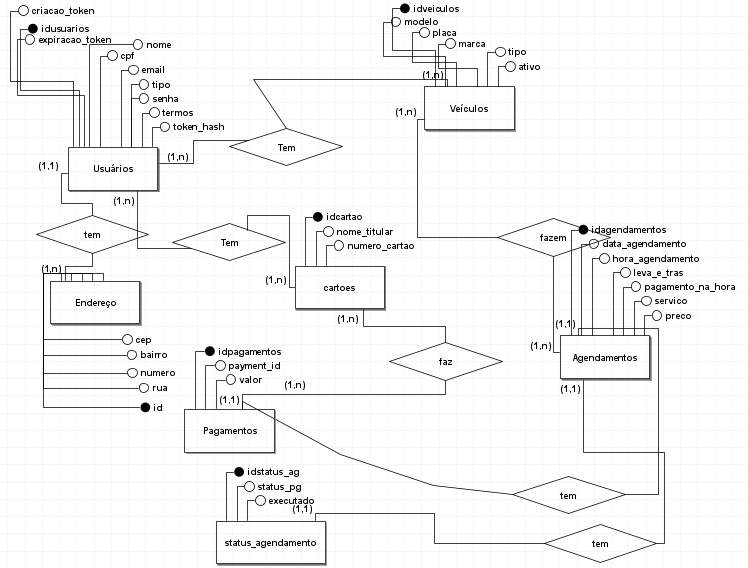
**Consultar Clientes (Administrador):**



## **3.3 MODELAGEM DE BANCO DE DADOS**

***3.3.1 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:

******

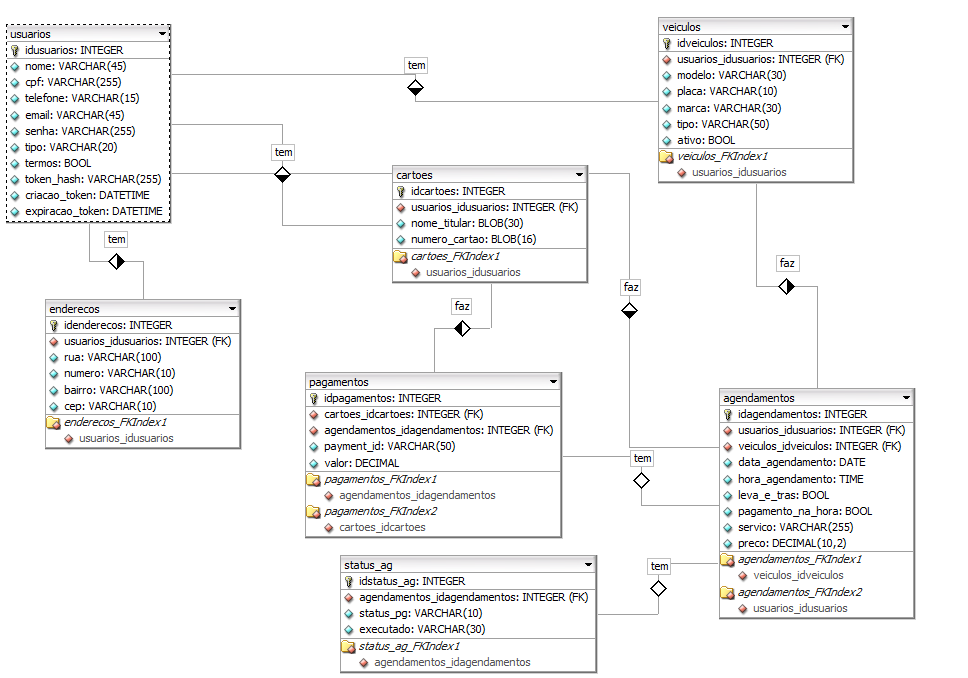
### **Relacionamentos**

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 **Catões**: 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).

***3.3.2 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\_idusarios** na tabela **enderecos** referencia a chave primária **idusarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus endereços.

## A chave estrangeira **usuarios\_idusarios** na tabela **veiculos** referencia a chave primária **idusarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus veículos.

## As chaves estrangeiras **usuarios\_idusarios** e **veiculos\_idveiculos** na tabela **agendamentos** referenciam, respectivamente, as chaves primárias **idusarios** 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\_idusarios** na tabela **enderecos** referencia a chave primária **idusarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus endereços.

## A chave estrangeira **usuarios\_idusarios** na tabela **veiculos** referencia a chave primária **idusarios** da tabela **usuarios**, implementando o relacionamento de um-para-muitos entre usuários e seus veículos.

## As chaves estrangeiras **usuarios\_idusarios** e **veiculos\_idveiculos** na tabela **agendamentos** referenciam, respectivamente, as chaves primárias **idusarios** 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).

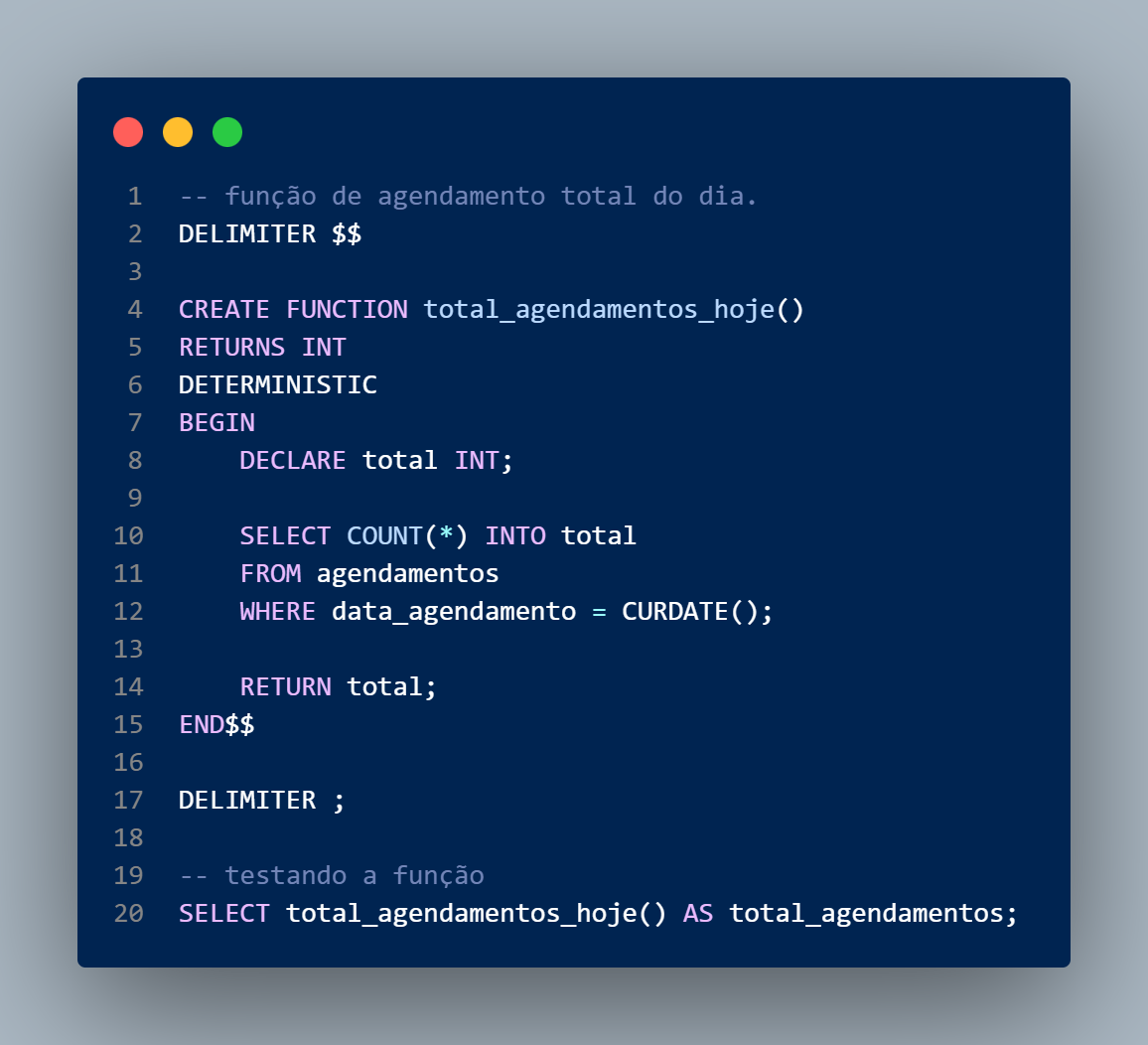
## A chave estrangeira **usuarios\_idusuarios** na tabela **cartoes** referencia a chave primária **idusuarios** da tabela **usuarios**, indicando a qual usuário o cartão pertence, (relacionamento um-para-N entre usuários e cartões).

## 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.

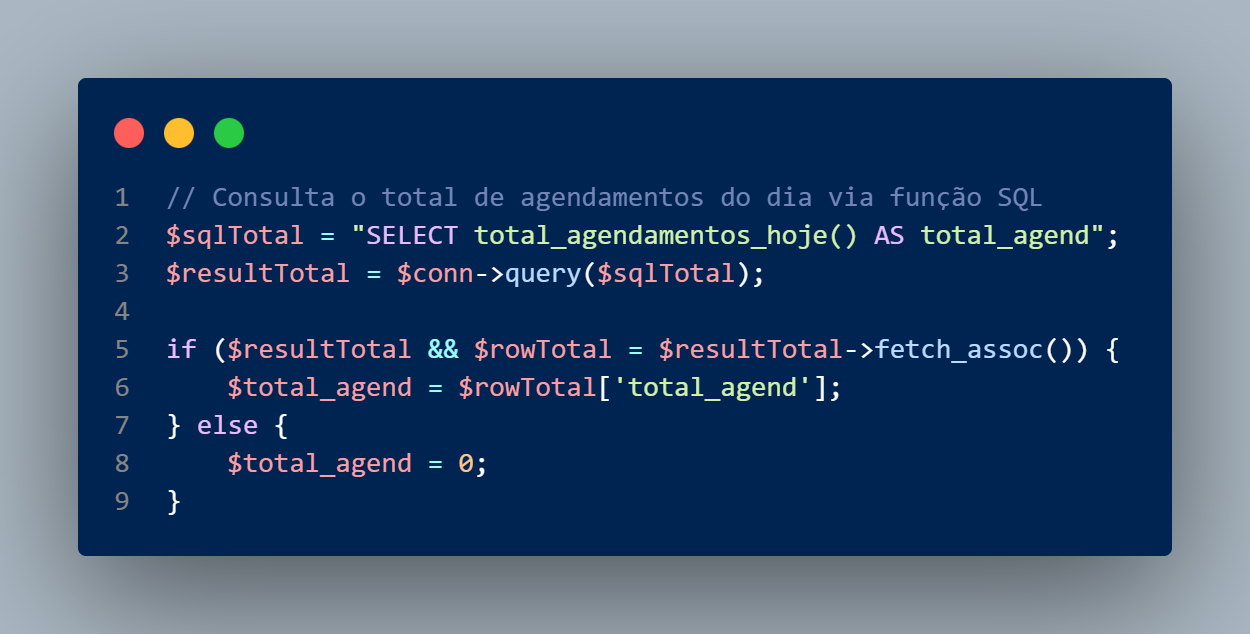
**3.4 RECURSOS**

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.

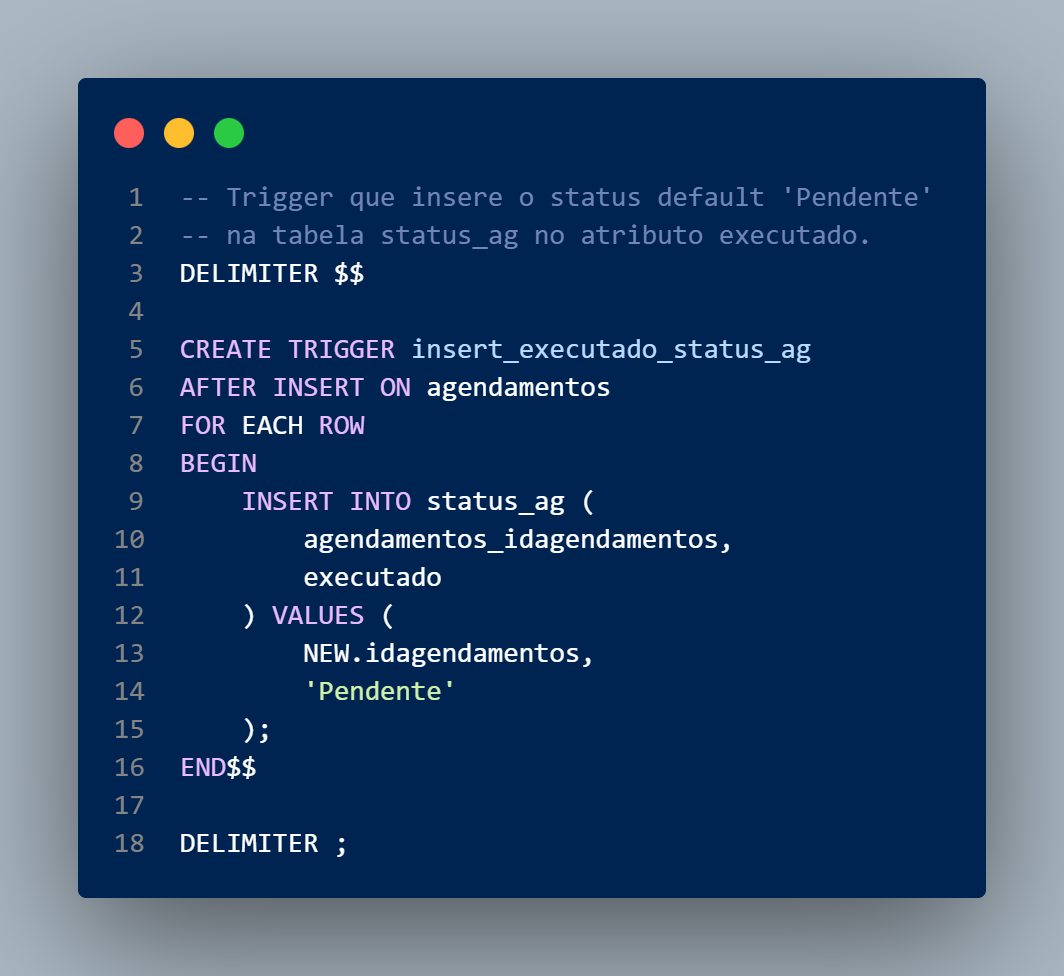
Código que gerou a function no banco de dados:

****

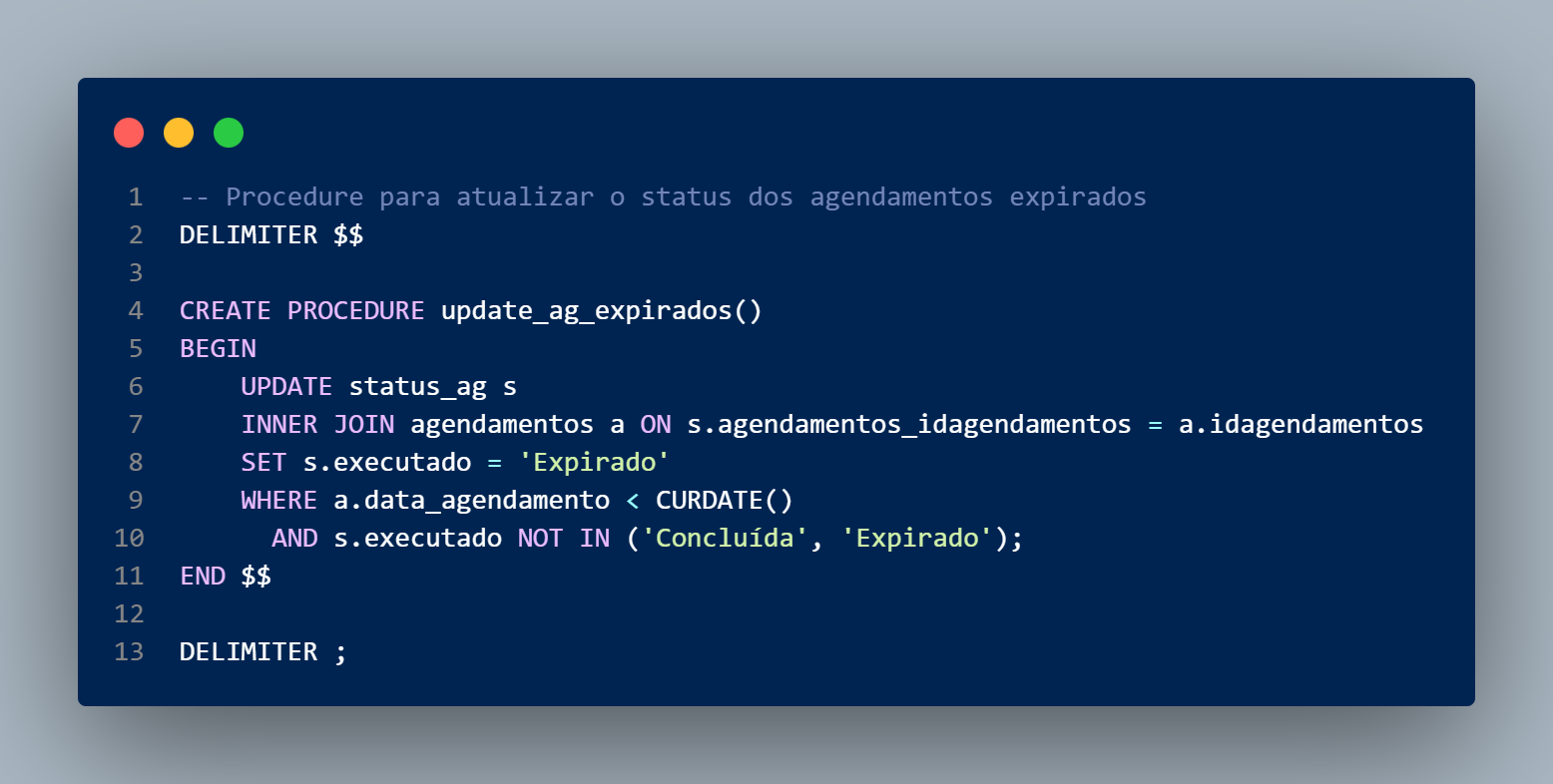
A função criada para a página de agendamentos diários do administrador, é executada no código PHP, da seguinte maneira:



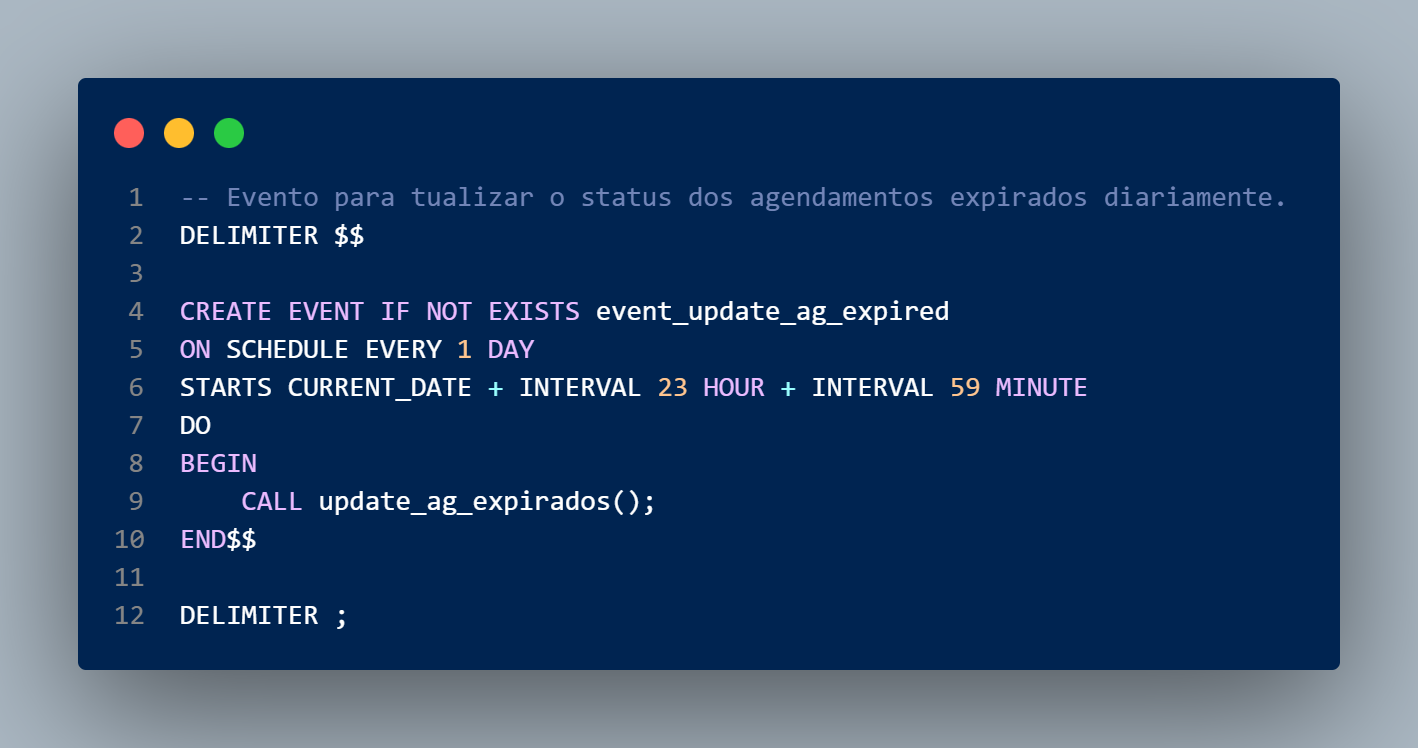
## Foi implementado o objeto Trigger: **insert\_executado\_status\_ag**, no banco de dados pois assim que um agendamento é criado a trigger insere o status no atributo executado na tabela status\_ag como default o dado ‘Pendente’, veja abaixo o código que criou a Trigger no banco de dados.



## O último objeto implementado no banco de dados foi uma procedure para atualizar status de agendamentos expirados: update\_ag\_expirados(), caso o cliente não apareceu. ou o agendamento ficou para trás sem execução manual do administrador, veja o código da procedure implementada:



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:



## **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, o[**Figma**](https://www.figma.com/design/PpyOYg0jN5SyXKjDlWTspq/Fatec-Projeto-Integrador-2%C2%BA-S?node-id=0-1&p=f&t=g2WxhTmHlvDLMuhr-0) e o [**Canvas**](https://www.google.com/search?gs_ssp=eJzj4tbP1TcwNMrIMyo3U2A0YHRg8GJNTswrSwQASccGEw&q=canva&oq=canva&gs_lcrp=EgZjaHJvbWUqEwgBEC4YgwEYxwEYsQMY0QMYgAQyBggAEEUYPDITCAEQLhiDARjHARixAxjRAxiABDINCAIQABiDARixAxiABDINCAMQABiDARixAxiABDINCAQQABiDARixAxiABDINCAUQABiDARixAxiABDIGCAYQRRg8MgYIBxBFGDzSAQgzMTM2ajBqN6gCALACAA&sourceid=chrome&ie=UTF-8).

## ***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:

Colocar as imagens aqui…

Imagens:

e aqui

Version [desk-top]:

colocar foto das telas versão desktop aqui

Version [desk-top] [light]:

e aqui

Version [mobile] [light]:

Colocar foto das telas versão mobile aqui

Version [mobile] [dark]:

e aqui

Nesta etapa, com o layout pronto, e validado pelo cliente, já é definido e pode seguir para a fase de desenvolvimento do software.

## ***4.3 Codificação***

## ***Ambiente de Desenvolvimento***

## Para o desenvolvimento do software será utilizada a IDE [**Visual Studio Code**](https://code.visualstudio.com), em sistema operacional Windows 10 / Windows 11 .

## ***Procedimentos de Desenvolvimento***

## Os Procedimentos do desenvolvimento do software será em etapas, sendo elas codificação do layout, funcionalidades requisitadas,...........

## ***Implementação*** ***Controle de versionamento (GitHub)***

## O versionamento da documentação, bem como do código fonte do software, será feito em repositório do [**GitHub**](https://github.com/Lucas-Ed/Pi-2-semestre), e será dividido em pastas, uma para documentação e outra para o software, após concluído a primeira versão terá um commit com o nome da versão concluída.

***Hospedagem***

A hospedagem do código fonte do software da API, será feito nos servidores da.

**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 Lava rápido seja apresentada de maneira eficiente, e funcional. Isso inclui o navegar entre as páginas ……………………….

***5.2 Resultado dos testes (Apresentação de resultados)***

***5.3 Garantia da qualidade (práticas adotadas)***

***5.4 Requisitos mínimos de hardware e software para usabilidade do sistema***

## **6. CONTRATO PARA DESENVOLVIMENTO DO SOFTWARE**

**7. PLANO DE EVOLUÇÃO**

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

* No Web site adicionar sessao de trabalhe conosco
* Foto de perfil do usuário
* Notificação para usuário
* Pagamento via gateway
* Admin editar serviços e os preços

A versão inicial visou cumprir o objetivo que era …………………….

**8. REFERÊNCIAS**

**Anexo I - Diário de bordo- físico**

**Anexo II - Cronograma efetivo**