CarDoctor

Sprint 3 – Porto Seguro

*Luisa Danielle - RM 555292*

*Michelle Potenza - RM 557702*

*Gustavo de Melo - RM 558595*

SUMÁRIO

[OBJETIVOS E JUSTIFICATIVAS DO PROJETO 2](#_Toc1699214339)

[DESCRIÇÃO DA UTILIZAÇÃO DA MATÉRIA 3](#_Toc1810183054)

[USUÁRIO 3](#_Toc522186202)

[TELEFONEUSUARIO 3](#_Toc543874008)

[LOGINUSUARIO 4](#_Toc2077206823)

[ENDERECOUSUARIO 4](#_Toc697136067)

[AUTOMOVELUSUARIO 4](#_Toc902719542)

[AUTOMOVELDIAGNÓSTICO 4](#_Toc1463266276)

[MANUTENCAOAUTOMOVEL 4](#_Toc1921936977)

[DAO 4](#_Toc1000108031)

[PROCEDIMENTOS 5](#_Toc2090888111)

[PROTÓTIPO 5](#_Toc683151484)

[BANCO DE DADOS 6](#_Toc767514286)

[USUÁRIO 6](#_Toc629851895)

[LOGINUSUARIO 6](#_Toc1186624139)

[ENDERECOUSUARIO 7](#_Toc1383446737)

[AUTOMOVELUSUARIO 7](#_Toc430960778)

[MODELO RELACIONAL 7](#_Toc24032142)

# OBJETIVOS E JUSTIFICATIVAS DO PROJETO

CarDoctor é uma Plataforma Digital que oferece a opção de realizar um Autodiagnostico do seu veículo sem a necessidade de ir até uma oficina mecânica e da dependência de um humano, visando a praticidade e podendo ser acessado de qualquer lugar do Brasil.

Atualmente muitas reclamações são feitas sobre os serviços da Porto Seguro devido a longas esperas, baixa confiança no diagnóstico do serviço do veículo oferecido, dificuldade na descrição detalhada do problema e tempo de entrega do veículo afetado devido ao manuseio dos processos.

Com a implementação da plataforma digital estes problemas recorrentes não aconteceram mais. Será possível que os clientes descrevam os problemas presentes em seu veículo e rapidamente seja gerado um diagnóstico e um orçamento justo visando as necessidades. Deste modo, facilitando tanto ao cliente quanto ao mecânico que já saberá brevemente o problema do veículo, podendo ter uma preparação necessária, assim agilizando o processo.

O principal objetivo é facilitar, simplificar e agilizar o processo de diagnóstico do problema do veículo e garantindo uma abordagem mais assertiva, ágil e sem ação humana, assim, promovendo uma melhor experiencia ao cliente e satisfação.

# DESCRIÇÃO DA UTILIZAÇÃO DA MATÉRIA

A linguagem de programação, Java, será utilizada para coletar os dados pessoais dos usuários, como nome, CPF, automóveis, telefones, endereços, feedbacks do sistema CARDOCTOR e entre outros. Além disso, a linguagem será responsável por estabelecer a conexão com o banco de dados da CARDOCTOR, viabilizando a execução das operações CRUD (Criar, Ler, Atualizar e Deletar), conforme especificado no projeto desenvolvido com essa linguagem. E atualmente, o código foi atualizado com a criação das APIs. Para melhor visualização das funcionalidades da matéria de Java no projeto, será anexado uma imagem com um diagrama com todas as classes desenvolvidas direcionadas para o projeto:

## USUÁRIO

PATH("usuario")

A classe tem como função declarar todas as informações do usuário. Além disso, nela existe um método para calcular a idade do usuário, de acordo com sua data de nascimento.

## TELEFONEUSUARIO

PATH("telefone")

A classe tem como objetivo declarar todas as informações pertinentes aos telefones dos usuários. Além disso, ela implementa dois métodos: o primeiro tem a função de formatar o valor referente ao número de telefone, utilizando a sintaxe de telefone no formato (XXXXX-XXXX); o segundo método é responsável por juntar alguns valores e transformar em uma String formatada na sintaxe de telefone (+DDI (DDD) NR\_TELEFONE) assegurando uma melhor visualização dos telefones dos usuários.

## LOGINUSUARIO

PATH("login")

A classe tem como objetivo declarar todas as informações pertinentes ao login cadastrado pelos usuários. Além disso, nessa classe foi aplicada uma restrição de mínimo 8 caracteres dentro do método set do atributo NR\_SENHA.

## ENDERECOUSUARIO

PATH("endereco")

A classe tem como objetivo declarar todas as informações pertinentes aos endereços dos usuários. Além disso, ela implementa dois métodos: o primeiro tem a função de formatar o valor referente ao CEP, utilizando a sintaxe de CEP no formato (XXXXX-XXX); o segundo método é responsável por juntar alguns valores e transformar em uma String formatada na sintaxe de logradouro (CEP, BAIRRO, CIDADE, ESTADO(SIGLA ESTADO)) assegurando uma melhor visualização dos endereços dos usuários.

## AUTOMOVELUSUARIO

PATH("automovel")

A classe tem como objetivo declarar todas as informações pertinentes relacionadas aos automóveis cadastrados pelo usuário. Além disso, ela implementa um método, sendo responsável por validar o ano do automóvel, assegurando que os anos futuros sejam identificados e, consequentemente, impedidos.

## AUTOMOVELDIAGNÓSTICO

PATH("diagnostico")

A classe tem como objetivo declarar todas as informações pertinentes ao auto diagnóstico digital realizado pelos usuários. Além disso, ela implementa dois métodos: o primeiro tem a função de formatar o valor referente ao horário, utilizando a sintaxe de horas no formato (HH:MM); o segundo método é responsável por validar a data do diagnóstico, assegurando que datas futuras sejam identificadas e, consequentemente, impedidas.

## MANUTENCAOAUTOMOVEL

PATH("manutencao")

A classe tem como objetivo declarar todas as informações pertinentes as manutenções realizadas pelos usuários em seus automóveis. Além disso, ela implementa um método, sendo responsável por validar a data da manutenção, assegurando que datas futuras sejam identificadas e, consequentemente, impedidas.

## DAO

As operações principais implementadas neste projeto são as seguintes:

1. **Adicionar**: Esta função é responsável por inserir novos registros na base de dados. Ela permite que dados adicionais sejam incluídos na tabela correspondente, expandindo o conjunto de informações disponíveis.
2. **Atualizar**: Esta função possibilita a modificação de registros existentes. Utilizando essa operação, é possível alterar os dados de um registro específico, garantindo que as informações na base de dados estejam sempre atualizadas e corretas.
3. **Deletar**: Esta função é utilizada para remover registros da base de dados. Ao executar esta operação, os registros selecionados são excluídos permanentemente, liberando espaço e mantendo a base de dados organizada.
4. **Consultar**: Embora não mencionado inicialmente, é essencial incluir a função de consulta (ou leitura), que permite recuperar e visualizar os dados armazenados na base de dados. Essa operação é fundamental para obter informações e gerar relatórios baseados nos registros existentes.

## RESOURCE

Cada classe possui as funções:

1. GET
2. GET (ID)
3. POST
4. PUT
5. DELETE



Link de acesso ao diagrama:

<https://lucid.app/lucidchart/143e3f30-c56a-4bdd-9439-340e4950dad2/edit?viewport_loc=-32%2C-28%2C2368%2C1232%2CHWEp-vi-RSFO&invitationId=inv_cceb89fe-ba3b-4bb4-a5ba-fce477c4507b>

## PROCEDIMENTOS

RM: 555292

SENHA: 070905

# BANCO DE DADOS

Restrições do banco de dados da CARDOCTOR:

## USUÁRIO

Usuario ADD CONSTRAINT chk\_nr\_cpf

CHECK (REGEXP\_LIKE(NR\_CPF, '^\d{3}\.\d{3}\.\d{3}\-\d{2}$')

OR REGEXP\_LIKE(NR\_CPF, '^\d{11}$'));

a restrição chk\_nr\_cpf garante que o valor do campo NR\_CPF siga um dos dois formatos válidos para CPF: com pontos e hífen ou apenas com números. Isso ajuda a garantir que os dados inseridos estejam no formato correto. A fim de gerar uma melhor visualização do elemento CPF dentro da tabela usuario.

## LOGINUSUARIO

ADD CONSTRAINT chk\_senha

CHECK (LENGTH(DS\_SENHA) >= 8)

LOGINUSUARIO

ADD CONSTRAINT chk\_email

CHECK (REGEXP\_LIKE(DS\_EMAIL, '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'));

Essas restrições ajudam a garantir a integridade e a validade dos dados inseridos na tabela, assegurando que as senhas sejam suficientemente longas e os e-mails estejam em um formato aceitável.

## ENDERECOUSUARIO

ADD CONSTRAINT chk\_nm\_estado

CHECK (NM\_ESTADO IN ('Acre', 'Alagoas', 'Amapá', 'Amazonas', 'Bahia', 'Ceará', 'Distrito Federal', 'Espírito Santo', 'Goiás', 'Maranhão', 'Mato Grosso', 'Mato Grosso do Sul', 'Minas Gerais', 'Pará', 'Paraíba', 'Paraná', 'Pernambuco', 'Piauí', 'Rio de Janeiro', 'Rio Grande do Norte', 'Rio Grande do Sul', 'Rondônia', 'Roraima', 'Santa Catarina', 'São Paulo', 'Sergipe', 'Tocantins'));

ADD CONSTRAINT chk\_sg\_estado

CHECK (SG\_ESTADO IN ('AC', 'AL', 'AP', 'AM', 'BA', 'CE', 'DF', 'ES', 'GO', 'MA', 'MT', 'MS', 'MG', 'PA', 'PB', 'PR', 'PE', 'PI', 'RJ', 'RN', 'RS', 'RO', 'RR', 'SC', 'SP', 'SE', 'TO'));

Essas restrições ajudam a manter a integridade dos dados, garantindo que apenas nomes completos e siglas válidas dos estados sejam registrados na tabela.

## AUTOMOVELUSUARIO

ADD CONSTRAINT chk\_placa\_formato

CHECK (REGEXP\_LIKE(DS\_PLACA, '^[A-Z]{3}[0-9][A-Z][0-9]{2}$'));

Essa restrição assegura que as placas inseridas no banco de dados estejam no formato correto e padronizado.

## MODELO RELACIONAL

