FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

CARLOS EDUARDO MENDONÇA DA SILVA

EDUARDO TOSHIO ROCHA OKUBO

KAUÊ ALEXANDRE DE OLIVEIRA

MATEUS VINICIUS DA CONCEIÇÃO SILVA

VITOR MACHADO MIRANDA

PEDRO HENRIQUE FIGUEIREDO DE OLIVEIRA

DOCUMENTAÇÃO DO PROJETO PORTO ASSISTANT

SÃO PAULO

2023

CARLOS EDUARDO MENDONÇA DA SILVA

EDUARDO TOSHIO ROCHA OKUBO

KAUÊ ALEXANDRE DE OLIVEIRA

MATEUS VINICIUS DA CONCEIÇÃO SILVA

VITOR MACHADO MIRANDA

PEDRO HENRIQUE FIGUEIREDO DE OLIVEIRA

DOCUMENTAÇÃO DO PROJETO PORTO ASSISTENT

Trabalho acadêmico apresentado à disciplina de Domain Driven Design do Curso Análise e Desenvolvimento de Sistemas da Faculdade de Informática e Administração Paulista como requisito de nota do Sprint 4 da Turma 1TDSPV. Requerido pelo prof. Leonardo Gasparini Romão.

SÃO PAULO

2023

SUMÁRIO

SÃO PAULO

2023

1 INTRODUÇÃO

Este estudo refere-se ao objetivo de sanar a dor, da falta de assertividade de modais para veículos pesados, que a Porto propôs para nós, da turma 1TDSPV da FIAP.

A seguradora Porto tem uma frota de guinchos para veículos pesados que são frequentemente utilizados para prestar serviços de assistência em rodovias e estradas. No entanto, muitas vezes esses guinchos não são assertivos o suficiente, o que pode causar atrasos e insatisfação por parte dos clientes. Com o objetivo de melhorar a qualidade do serviço prestado, nós da PortoAssistant pretendemos desenvolver um projeto de programação que permita aprimorar a assertividade dos guinchos para veículos pesados dessa seguradora.

Para alcançar esse objetivo, o projeto será dividido em várias etapas, sendo elas, a parte de Cadastro, onde haverá a coleta de dados pessoais do cliente e dados pertinentes ao veículo pesado; a parte de interação com Inteligência Artificial no nosso site/aplicativo com o intuito de diminuir a mão-de-obra humana, evitando gastos e falhas humanas.

O resultado desse projeto será uma melhoria significativa na qualidade do serviço, sobre o modal, prestado pela seguradora Porto, que poderá ser traduzida em uma maior satisfação dos clientes e uma maior fidelização deles.

2 NOSSA SOLUÇÃO

Coma empresa contratada para solucionar essa dor, uma das nossas principais preocupações é oferecer aos nossos clientes serviços de qualidade e confiabilidade. No setor de transporte de veículos pesados, um dos problemas mais recorrentes é a falta de assertividade dos guinchos em situações de resgate ou transporte de veículos acidentados.

Para resolver esse problema, propomos o desenvolvimento de um projeto de programação que permita o uso de tecnologia avançada para melhorar a eficiência e assertividade dos guinchos em situações de resgate de veículos pesados. O projeto consistirá em um sistema de coleta de informações que são essenciais, no momento do cadastro, sendo informações como: Nome Completo, CPF, RG, CNH, Telefone, Gênero, Email, Senha, Modelo do veículo, Peso, Altura, Largura, Comprimento, Quantidade de Eixos, Tipo de Eixo e Tipo de Chassi. No momento de uso do nosso serviço, o cliente irá acessar a aba “Atendimento” e começará enviando uma mensagem ao nosso ChatBot, que irá guiar o processo de coleta de informações sobre o problema, começando pela confirmação dos dados do veículo, cadastrado no sistema, se possuem alguma alteração. Caso houver, pedirá ao cliente, que confirme o novo dado do veículo. Caso não houver, prosseguirá com o atendimento, perguntando se o veículo está transportando carga. Se estiver, exigirá ao cliente, que comunique a empresa do transporte para o transbordo da carga. Se não estiver, prosseguirá realizando algumas perguntas sobre o estado do veículo, por exemplo “O veículo apresenta fumaças aparentes?”, “O pneu está furado?”, “Consegue dar partida no veículo?”, exigindo o endereço do local etc. Isso nos fornecerá informações para analisar se o problema necessitará de um guincho para reboque do veículo, ou se apenas um funcionário resolverá o problema, podendo ser elétrico ou mecânico de fácil manutenção.

Para garantir o sucesso do projeto, será necessário contar com uma equipe multidisciplinar de desenvolvedores, analistas de dados e especialistas em logística e transportes. Além disso, será necessário investir em treinamento e capacitação dos nossos operadores para garantir que eles possam utilizar o sistema com eficiência e precisão.

Com a implementação desse projeto, esperamos melhorar significativamente a qualidade e confiabilidade dos nossos serviços de resgate e transporte de veículos pesados, proporcionando aos nossos clientes maior tranquilidade e segurança em emergências.

3 COMO FAZER CHAMADAS A API

Como requisição da matéria de Domain Drive Design, foi desenvolvido uma API RESTful na linguagem Java usando o modelo de arquitetura JAX-RS. A aplicação foi pensada para que possa integrar o Front-End com o banco de dados, tornando possível a interação com os dados armazenados.

A API desenvolvida possui 8 endpoints que possibilitam operações de CRUD com as entidades do banco de dados, ou seja, cada um desses endpoints possuem os métodos GET (Um para trazer todos os dados da respectiva tabela e outro para trazer um específico por meio do seu ID), POST, PUT e DELETE.

Os endpoints podem ser usados para interagir com as seguintes entidades do banco de dados:

Entidade	Endpoint	Sobre
Chamada	“/chamada”	Recupera e manipula informações das chamadas dos clientes.
Cliente	“/cliente”	Recupera e manipula informações dos clientes da seguradora.
Colaborador	“/colaborador”	Recupera e manipula informações dos colaboradores da seguradora.
Medida	“/medida”	Recupera e manipula informações de medidas dos veículos e modais.
Modal do Colaborador	“/modal-colaborador”	Recupera e manipula as ligações dos modais com seus respectivos colaboradores.
Modal	“/modal”	Recupera e manipula informações dos modais.
Veículo do Cliente	“/veiculo-cliente”	Recupera e manipula ligações dos veículos com seus respectivos proprietários.
Veículo	“/veiculo”	Recupera e manipula informações dos veículos.

Como fazer a chamada?

Com o programa rodando, basta usar a seguinte o URL para acessar o endpoint desejado:

http://localhost:8081/api/{endpoint}

Exemplos de Chamadas da API

Recuperando todos os dados

Para recuperar todos os dados presentes na tabela do banco basta informar o endpoint ao qual você deseja acessar as informações e realizar uma requisição GET. Você receberá como retorno um JSON com os dados dos clientes.

URL para a chamada:

http://localhost:8081/api/veiculo

Retorno:

```
[
    {
        "ano_veiculo": 2020,
        "apolice_veiculo": 45695816365,
        "id_medida": {
            "id": 1,
            ...
        },
        "id_veiculo": 1,
        "modelo_veiculo": "IDK231",
        "nr_chassi": "asd2j7jij7aw1",
        "placa_veiculo": "FAW9020",
        ...
    },
{
        "ano_veiculo": 2013,
        "apolice_veiculo": 78542389765,
        "id_medida": {
            "id": 2,
            ...
        },
        "id_veiculo": 2,
        "modelo_veiculo": "R-440",
        "nr_chassi": "sgt3342233r3r",
        "placa_veiculo": "KBM8653",
        ...
    }
]
```

Status HTTP:

200 OK	Retornado com sucesso.
404 NOT FOUND	Nenhum dado encontrado.

Recuperando um dado específico

Além do endpoint, adicione também o ID da informação que você deseja recuperar na URL e realize uma requisição GET, assim você terá como retorno um único objeto JSON do dado com respectivo ID.

URL para a chamada:

http://localhost:8081/api/veiculo/1

Retorno:


```
[
    {
        "ano_veiculo": 2020,
        "apolice_veiculo": 45695816365,
        "id_medida": {
            "altura": 5.0,
            "comprimento": 12.0,
            "id": 1,
            "largura": 6.0,
            "peso": 12.0,
            "peso_suportado": 32.0
        },
        "id_veiculo": 1,
        "modelo_veiculo": "IDK231",
        "nr_chassi": "asd2j7jij7aw1",
        "placa_veiculo": "FAW9020",
        "qtd_eixos_veiculo": 3,
        "renavan_veiculo": 19238192893,
        "tp_chassi": "monobloco",
        "tp_eixo": "duplo"
    }
]
```


Status HTTP:

200 OK	Retornado com sucesso.
404 NOT FOUND	Nenhum dado encontrado.

Adicionando uma nova informação

Para adicionar uma nova informação você precisará enviar, por meio de uma requisição POST, um objeto JSON com os atributos da entidade, respeitando as regras e restrições do banco de dados, no endpoint respectivo a tabela. Não é necessário passar o ID, pois o banco se encarregará de colocá-lo.

URL para a chamada:

http://localhost:8081/api/veiculo

Exemplo de objeto que deve ser enviado:

```
[
    {
        "ano_veiculo": 2017,
        "apolice_veiculo": 86943216364,
        "id_medida": {
            "id": 3
        },
        "modelo_veiculo": "Atego 2426",
        "nr_chassi": "fhdu8394fjs93",
        "placa_veiculo": "SFY0653",
        "qtd_eixos_veiculo": 3,
        "renavan_veiculo": 98534551893,
        "tp_chassi": "Chassi Truck",
        "tp_eixo": "duplo"
    }
]
```

Status HTTP:

201 CREATED	Criado com sucesso.
400 BAD REQUEST	Envio incorreto.

Atualizando uma informação

Caso precise atualizar os dados é possível usar uma requisição PUT e usar a URL com endpoint e o ID da informação que você deseja alterar. Você também precisará enviar um objeto JSON com todos os atributos da entidade, alterando aqueles que devem ser atualizados. Não é necessário enviar o ID, pois ele será recuperado da URL pela própria API.

URL para a chamada:

http://localhost:8081/api/veiculo/1

Exemplo de objeto que deve ser enviado:

```
[
    {
        "ano_veiculo": 2018,
        "apolice_veiculo": 86943216364,
        "id_medida": {
            "id": 3
        },
        "modelo_veiculo": "Atego 2426",
        "nr_chassi": "fhdu8394fjs93",
        "placa_veiculo": "GJK49832",
        "qtd_eixos_veiculo": 3,
        "renavan_veiculo": 98534551893,
        "tp_chassi": "Chassi Truck",
        "tp_eixo": "duplo"
    }
]
```

Status HTTP:

200 OK	Atualizado com sucesso.
404 NOT FOUND	Nenhum dado encontrado.

Deletando uma informação

O método DELETE necessita apenas de informar o endpoint e o ID do dado que se deseja deletar na URL e a informação será apagada no banco de dados.

URL para chamada:

http://localhost:8081/api/veiculo/3

Status HTTP:

200 OK	Deletado com sucesso.
404 NOT FOUND	Nenhum dado encontrado.

Verificação de Login

A API ainda possui dois endpoint para fazer verificação do login do cliente e do colaborador. Essa verificação de login pode ser acessada nessas URLs:

http://localhost:8081/api/cliente/login
http://localhost:8081/api/colaborardor/login

Os dados podem ser recuperados de um formulário no Front-End e enviados em um JSON com a seguinte estrutura:

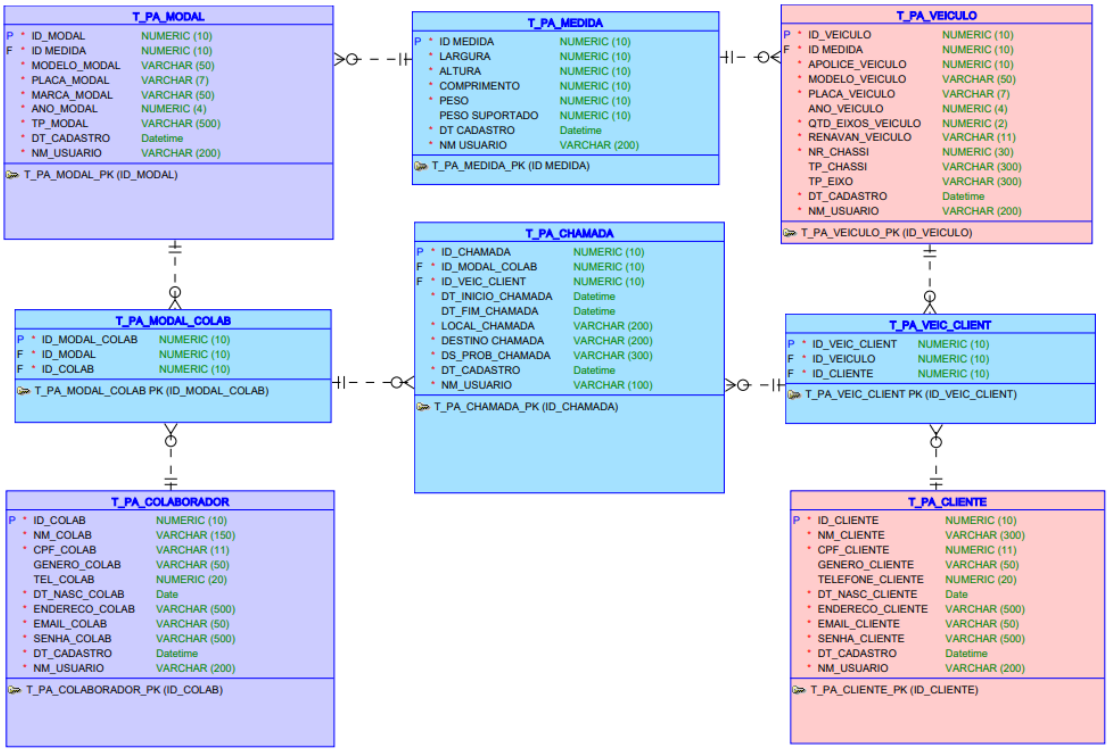
```
{
        "email_cliente": "exemplo@email.com",
         "senha_cliente": "ExemploDeSenha",
}
```

O e-mail será verificado no banco de dados, no caso de existir a senha será verificada. Se tudo estiver de acordo, será retornado um JSON com os dados desse cliente ou colaborador.

Status HTTP:

202 ACCEPTED	Retornado com sucesso.
404 NOT FOUND	Email não encontrado.
406 NOT ACCEPTABLE	Senha incorreta.

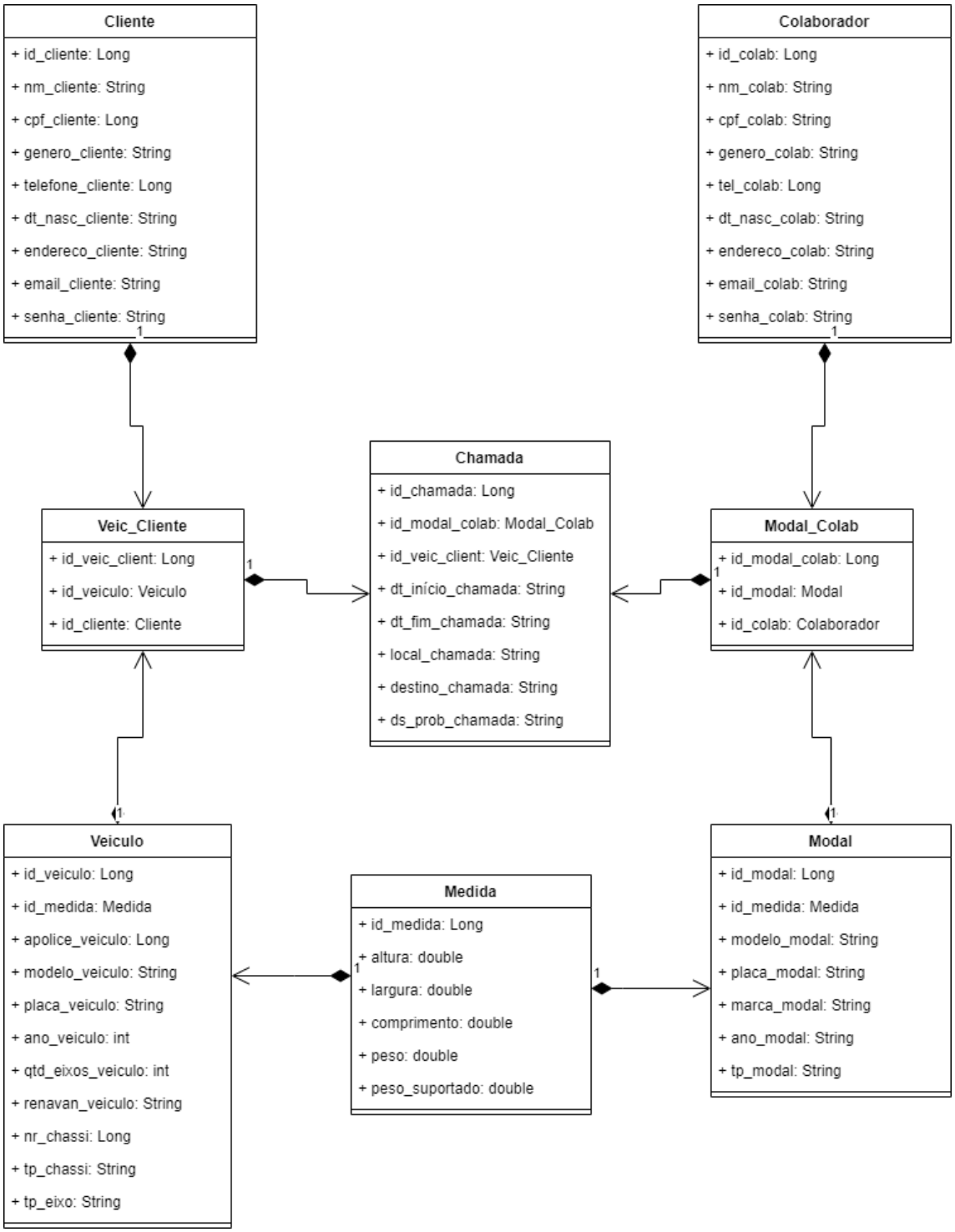
4 MODELO DO BANCO DE DADOS



SÃO PAULO

2023

5 UML DAS CLASSES



SÃO PAULO

2023