

ALEJANDRO RABELO – 93606

CRISTIAN ALVARO CONDORI PAUCARA - 550509

DIEGO SEITI OGITA IACABO – 551289

JOÃO LUCAS MARQUES COSTA - 98376

LUIZ FELIPE AZEVEDO DE OLIVEIRA - 550348

Projeto Bike Segura

São Paulo, 2023

ALEJANDRO RABELO – 93606

CRISTIAN ALVARO CONDORI PAUCARA - 550509

DIEGO SEITI OGITA IACABO – 551289

JOÃO LUCAS MARQUES COSTA - 98376

LUIZ FELIPE AZEVEDO DE OLIVEIRA - 550348

Projeto Bike Segura

Trabalho de aproveitamento de semestre solicitado pelo professor João Carlos Lima e Silva como um dos pré-requisitos para a avaliação da disciplina de Domain Driven Design.

São Paulo, 2023

Sumário

DESCRIÇÃO DA SOLUÇÃO.....	2
PRINCIPAIS FUNCIONALIDADES	3
TABELA DE ENDPOINTS	4
PROTÓTIPO	5
PROCEDIMENTO PARA RODAR APLICAÇÃO	7
MODELAGEM DO BANCO DE DADOS.....	7

DESCRIÇÃO DA SOLUÇÃO

Analisando e compreendendo os problemas presentes no procedimento atual e as necessidades que precisam ser atendidas para uma solução satisfatória, o grupo se prontificou a desenvolver um projeto focado na simplicidade e praticidade do atendimento ao cliente, possibilitando que a empresa abranja um público maior em suas vistorias, com mais velocidade e segurança.

Dessa forma, com o objetivo de automatizar todo o processo de vistoria e validação, foi pensado um sistema que seja capaz de receber imagens e, através de uma inteligência artificial, como, por exemplo, a Teachable Machine, da Google, analisar se esses arquivos estão de acordo com os padrões e critérios estabelecidos pela empresa para a aceitação do produto a ser segurado. Ainda no tópico das imagens, outro ponto relevante é a presença de um mecanismo que impossibilite o acesso de fotos da galeria, obrigando o usuário a tirá-las apenas no momento da vistoria.

Buscando uma precisão ainda maior, esse procedimento foi organizado em etapas. Sendo assim, há a divisão da inserção de imagens em 6 campos, cada um destinado a partes específicas da bicicleta: um campo para o guidão, outro para o banco, número de série, corrente e pedal, acessórios e um para a bike inteira. Seguindo essa linha, acreditamos que o processo de validação se tornará mais assertivo, aumentando a detecção de possíveis avarias e incoerências e diminuindo o número de fraudes.

Pela limitação de conhecimento com a manipulação de imagens dentro de Java, as principais funcionalidades ficaram com a captação e inserção de dados como o cadastro de cliente e bicicleta. As classes de vistoria e validação, que envolvem imagens e implementação de IA ainda não estão funcionais.

PRINCIPAIS FUNCIONALIDADES

Listar

Método: GET

Retorna a lista de todos os dados por ID

Produz: Resposta em formato JSON contendo a lista do endpoint.

Consultar por ID:

Método: GET

Retorna os detalhes de um endpoint específico com base no ID fornecido.

Produz: Resposta em formato JSON contendo os detalhes da linha dados do endpoint.

Cadastrar

Método: POST

Cadastra uma nova classe

Consumir: Recebe dados no corpo da requisição em formato JSON.

Produz: Resposta indicando o sucesso da operação.

Deletar por ID:

Método: DELETE

Deleta uma insert com base no ID fornecido.

Produz: Resposta indicando o sucesso da operação. Retorna um status 200 (OK) se os dados foram removidos com sucesso, ou um status 404 (NOT FOUND) se os dados não foram encontrados.

Atualizar:

Método: PUT (Somente em Clientes e Bicicletas)

Atualiza uma insert com base no ID fornecido.

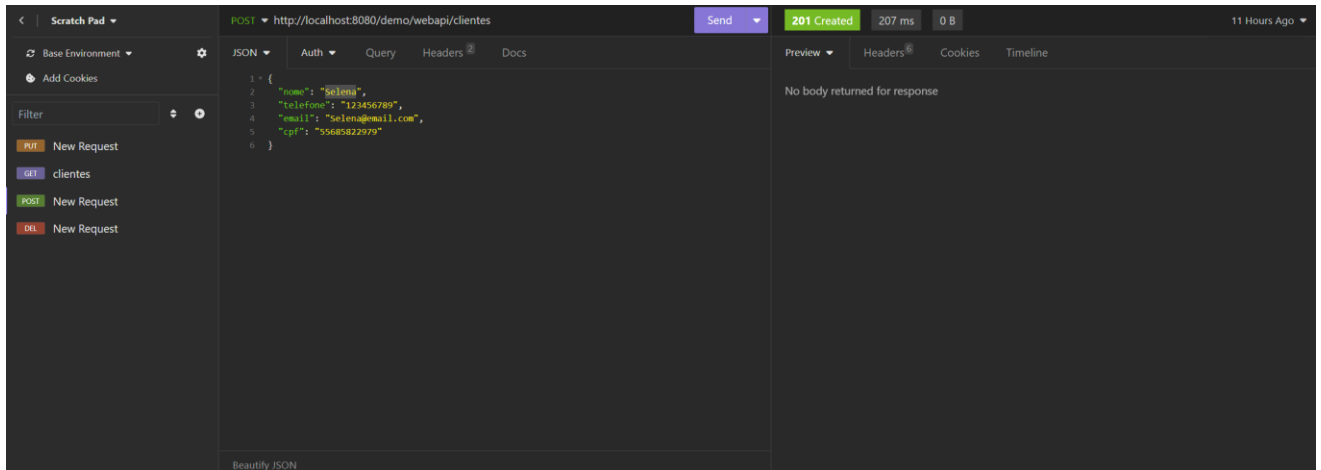
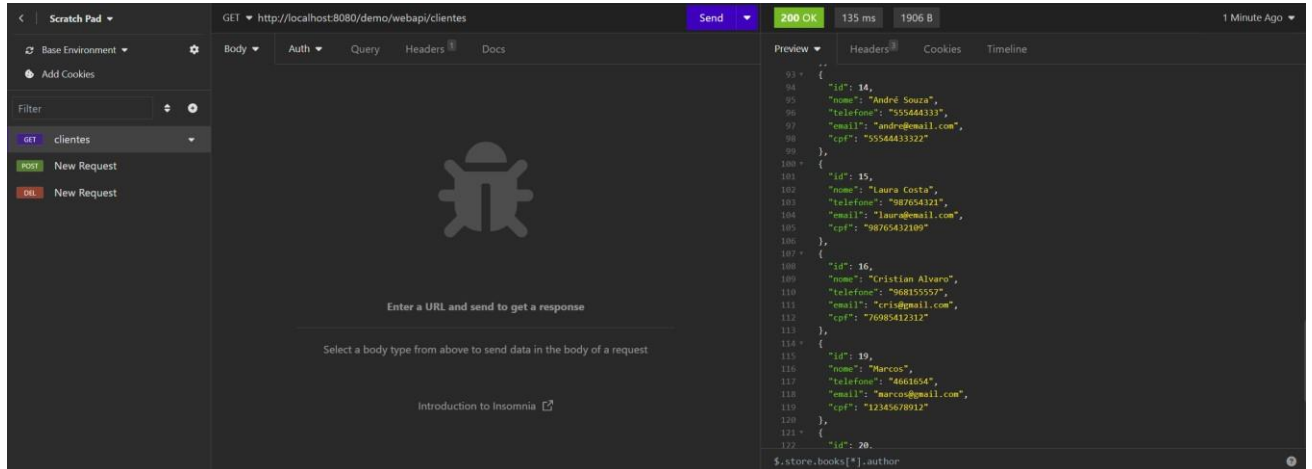
Produz: Resposta indicando o sucesso da operação. Retorna um status 200 (OK) se os dados foram removidos com sucesso, ou um status 404 (NOT FOUND) se os dados não foram encontrados.

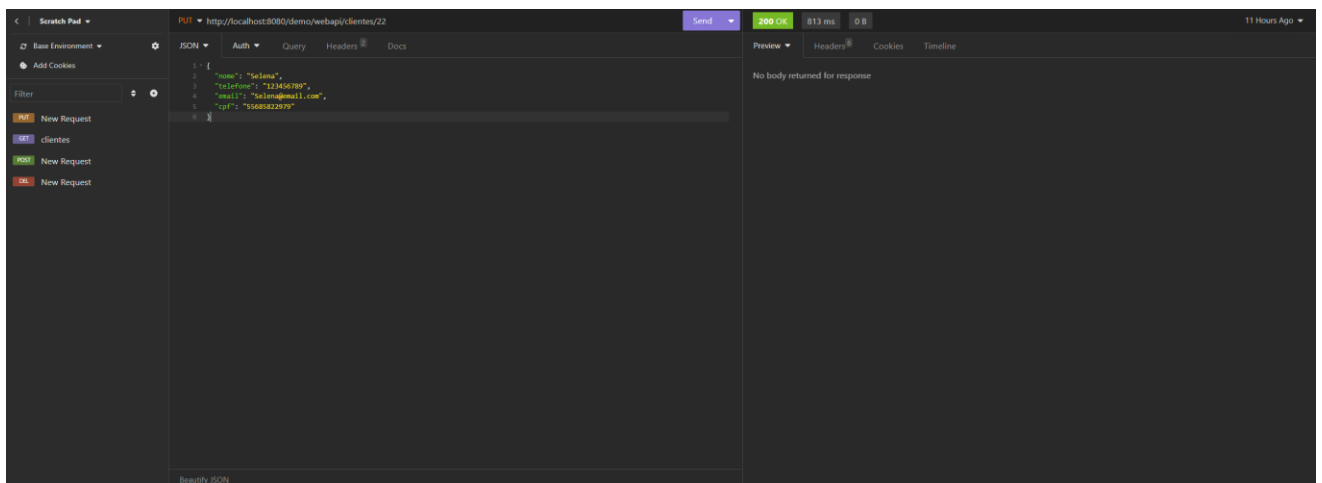
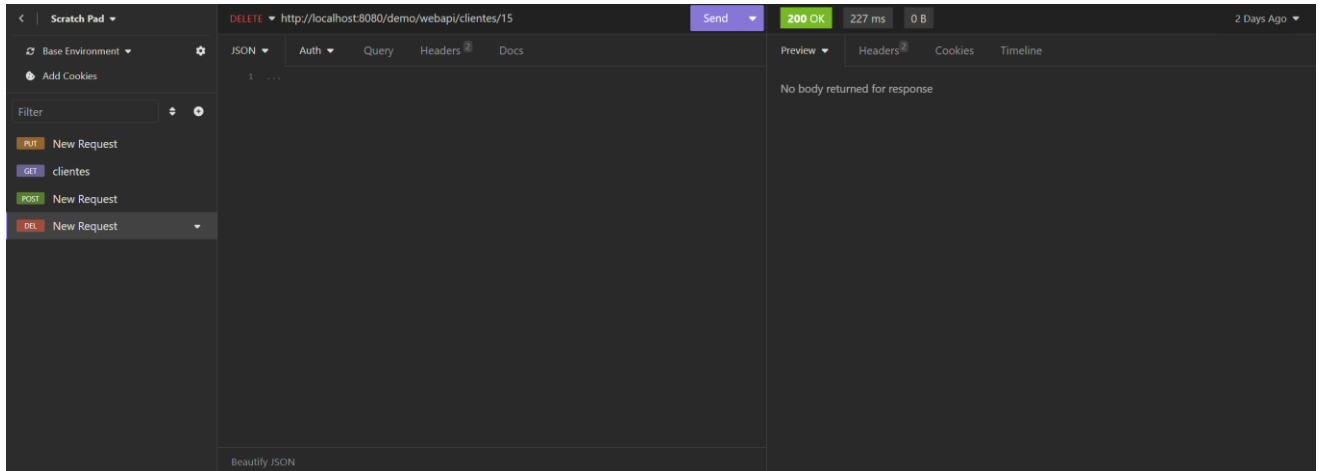
TABELA DE ENDPOINTS

Link dos endpoints:

- Clientes: "http://localhost:8080/demo/webapi/clientes"
- Bicicletas: "http://localhost:8080/demo/webapi/bicicletas"
- Vistorias: "http://localhost:8080/demo/webapi/vistorias"
- Sinistros: "http://localhost:8080/demo/webapi/sinistros"

PROTÓTIPO





PROCEDIMENTO PARA RODAR APLICAÇÃO

MODELAGEM DO BANCO DE DADOS

