

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

FIAP

DANIEL STUART – RM550897

GABRIEL NEVES – RM552244 – (1TDPSM)

GABRIEL SAMPAIO – RM552342

RAFAEL MENDONÇA – RM552422 – (1TDPSM)

RENATA BELILA – RM552315 – (1TDPSG)

Software Design & TX - 4ª Entrega

Documentação do projeto

SÃO PAULO - SP

2023

DANIEL STUART – RM550897

GABRIEL NEVES – RM552244 – (1TDPSM)

GABRIEL SAMPAIO – RM552342

RAFAEL MENDONÇA – RM552422 – (1TDPSM)

RENATA BELILA – RM552315 – (1TDPSG)

Software Design & TX - 4ª Entrega

Documentação do projeto da 4ª Sprint referente ao Challenge da Porto

Projeto de pesquisa IV, submetido à faculdade de informática e administração paulista (FIAP), como requisito necessário e condição obrigatória para a conclusão do semestre do Curso de Análise e Desenvolvimento de Sistemas (ADS).

**Orientador(a)**: Prof. Dr. Sidney Ferreira Leite

SÃO PAULO - SP

2023

### RESUMO

Este projeto aborda a implementação de uma vistoria automatizada para seguros de bicicletas, visando proporcionar praticidade e segurança aos clientes. Ao adotar uma abordagem baseada em inteligência artificial (IA) e tecnologias de reconhecimento visual, a solução elimina a necessidade de vistorias manuais em seguradoras, simplificando o processo para os usuários. Além disso, durante a vistoria, uma ligação entre o proprietário e o seguro será oficializada por meio do número de série da bicicleta, geralmente encontrada no chassi da mesma, assim fazendo com que o cliente acabe por se tornar o dono legal da bicicleta, da qual também, está sendo usada para gerar o seguro, logo reduzindo fraudes potenciais.

O projeto busca identificar possíveis problemas nas bicicletas por meio de inspeções automatizadas utilizando de uma IA, câmeras, sensores e algoritmos de aprendizado de máquina. A IA guiará os usuários durante o processo de vistoria, solicitando vídeos e fotos da bicicleta, além de Informações como notas fiscais e registros de peças que foram substituídas devem ser anexadas para documentação precisa.

Os resultados esperados incluem agilização da vistoria, redução de custos, prevenção contra fraudes e análise precisa de danos por meio de algoritmos de aprendizado de máquina da qual irá ler uma imagem e analisá-la a partir de uma biblioteca OCR. A vistoria automatizada busca detectar amassados, arranhões e rachaduras, avaliar danos e estimar custos de reparação. Além disso, a tecnologia contribui para a detecção de tentativas de fraude, comparando imagens atuais com vistorias anteriores.

Através da integração da IA, reconhecimento visual e documentação completa, o projeto visa melhorar a experiência dos usuários e aumentar a segurança das bicicletas, abordando questões de segurança e necessidades dos ciclistas modernos. A finalização da vistoria resulta na emissão do seguro, após confirmação dos dados e aprovação.

**Palavras-Chave: Vistoria automatizada, seguro de bicicleta, inteligência artificial, reconhecimento visual, prevenção de fraudes.**

### ABSTRACT

This project addresses the implementation of an automated inspection system for bicycle insurance, aiming to provide convenience and security to clients. By adopting an approach based on artificial intelligence (AI) and visual recognition technologies, the solution eliminates the need for manual inspections at insurance companies, simplifying the process for users. Additionally, during the inspection, a link between the owner and the insurance will be formalized through the bicycle's serial number, thus making the customer the legal owner of the bicycle for which the insurance is being generated, consequently reducing potential fraud.

The project seeks to identify possible issues in bicycles through automated inspections using AI, cameras, sensors, and machine learning algorithms. The AI guides users during the inspection process, requesting videos and photos of the bicycle, along with information such as invoices and records of replaced parts that need to be attached for accurate documentation.

Expected outcomes include streamlining inspections, cost reduction, fraud prevention, and precise damage analysis through machine learning algorithms that will read an image and analyze it using an OCR library. The automated inspection aims to detect dents, scratches, and cracks, assess damages, and estimate repair costs. Moreover, the technology contributes to fraud detection by comparing current images with past inspections.

Through the integration of AI, visual recognition, and comprehensive documentation, the project aims to enhance user experience and increase bicycle security, addressing safety concerns and modern cyclist needs. The completion of the inspection process results in the issuance of insurance after data confirmation and approval.

**Key-words: Automated inspection, bicycle insurance, artificial intelligence, visual recognition, fraud prevention.**

### SUMÁRIO

[1. EXPLICAÇÃO DO PROJETO 5](#_Toc14367)

[1.1. JUSTIFICATIVA 5](#_Toc14368)

[1.2. OBJETIVO 6](#_Toc14369)

[1.3. CONCLUSÃO 7](#_Toc14370)

# 1. EXPLICAÇÃO DO PROJETO

## 1.1.JUSTIFICATIVA

Um sistema de validação online de seguros de bicicletas automatizado e robotizado é importante porque pode melhorar significativamente a eficiência e a precisão do processo de validação e reduzir o tempo necessário para o processamento de reclamações. Além disso, um sistema online pode ser acessado a qualquer momento e de qualquer lugar, o que é conveniente tanto para o consumidor quanto para a seguradora.

Atualmente, muitas seguradoras de bicicletas dependem de intervenção humana para validar as reclamações de roubo ou furto de bicicletas. Isso pode levar a atrasos no processamento das reclamações, além de aumentar a probabilidade de erros humanos. Além disso, muitos consumidores acham o processo de validação de seguros muito burocrático e demorado.

De acordo com um estudo da Accenture, em 2020, mais de 80% dos consumidores esperam que as empresas usem tecnologias como inteligência artificial e automação para melhorar a experiência do cliente. Além disso, 60% dos consumidores esperam que as empresas usem tecnologias digitais para tornar a experiência de compra mais conveniente.

Com um sistema de validação online de seguros de bicicletas robotizado, as seguradoras podem validar as reclamações de forma mais rápida e precisa. Um sistema automatizado pode comparar fotos de bicicletas registradas pelos consumidores com fotos da bicicleta roubada ou furtada para confirmar a propriedade. Isso pode levar a uma redução nos prêmios pagos pelos consumidores, pois as seguradoras podem detectar fraudes com mais eficiência e precisão.

Além disso, um sistema online pode ser acessado a qualquer momento e de qualquer lugar, o que é conveniente tanto para o consumidor quanto para a seguradora. Isso pode ajudar a melhorar a experiência do cliente e aumentar a fidelidade do consumidor.

Em resumo, um sistema de validação online de seguros de bicicletas robotizado é importante porque pode melhorar a eficiência e a precisão do processo de validação, reduzir o tempo necessário para o processamento de reclamações, aumentar a conveniência para o consumidor e reduzir o risco de erros humanos.

## 1.2.OBJETIVO

Nossa empresa tem como objetivo criar um sistema de vistoria online para a Porto Seguro sem a necessidade de intervenção humana, visando praticidade, facilidade e economia de tempo. Visamos entregar um sistema moderno e seguro utilizando as tecnologias mais atuais do mercado.

Nossa solução consiste na utilização TensorFlow, uma biblioteca do React de código aberto usada para o aprendizado de máquinas que pode ser usada em diversos setores e tarefas. Para realizar o treinamento dessa biblioteca usamos o Coco, um Dataset onde obtemos imagens de bicicletas reais e usamos em nosso projeto objetivando nossa I.A a ficar cada vez mais inteligente e menos propensa a erros. Temos como foco impedir vistorias fraudulentas com imagens de bicicletas pegas da internet ou imagens espelhadas escondendo possíveis arranhões ou amassados. Para isso foi usado um sistema capaz de reconhecer imagens modificadas, espelhadas e/ou baixadas da internet.

Ademais, não haveria gasto para alimentar o banco de dados com imagens de bikes, pois com algumas modificações, o próprio cliente, no momento da vistoria da bicicleta, iria apresentar a foto de sua bicicleta, treinando a I.A automaticamente, a possibilitando de aprender com seus próprios erros aumentando sua eficiência a cada vistoria realizada, tudo isso sempre com uma interface intuitiva e de fácil acesso para todos, onde mostraremos mensagens avisando o cliente de qual parte da bicicleta precisamos e mensagens de aprovação ou reprovação.

Além disso, um dos nossos diferenciais está na volatilidade do nosso sistema, pois é de fácil modificação, o que permite a utilização dele em inúmeros processos de vistorias podendo ser encaixado seja em vistoria de veículos, casas ou até mesmo de celulares. Cenários diversos que podem ser resolvidos com apenas um sistema de fácil utilização e tecnologia inovadoras.

Portanto, utilizado o sistema de vistoria BikeIsure, a Porto não só conseguirá ofertar seus seguros com um menor gasto, como também passará na frente da concorrência, oferecendo um seguro de forma rápida e segura para seus clientes, chamando a atenção de novos consumidores com agilidade e praticidade.

## 1.3.CONCLUSÃO

Atualmente o nosso sistema se encontra no estágio final. Já é possível reconhecer números de séries e imagens fraudulentas que ao decorrer do ano era nossa maior dificuldade para a conclusão do projeto, conseguimos realizar tais funcionalidades devido as tecnologias de alta nível e modernidade.

Uma de nossas metas ao finalizar o projeto seria simplicidade que os clientes encontrariam para usar o sistema, se tornando democrático para todo que quiserem um seguro, desde usuários que já tivessem facilidade com a tecnologia, até as pessoas que não são tão acostumadas ao mundo digital.

Além disso, nós também queríamos construir algo que trouxesse uma proximidade da Porto com os seus clientes, além de gerar economia e segurança para a empresa, logo, criamos o sistema onde o próprio consumidor da Porto é o responsável por alimentar o sistema de banco de dados da empresa com fotos de suas bicicletas. Essa prática gerou uma dependência da I.A pois agora ela consegue se autocorrigir, ou seja, aprender com seus próprios erros.

Ademais, é válido frisar sobre a versatilidade do nosso projeto, capaz de realizar a vistoria não só de bicicletas, como também de outros seguros ofertados pelo Porto.

Para finalizar, nosso sistema possui atualmente o que os consumidores mais querem: agilidade e simplicidade. Capaz de realizar em minutos o que poderia levar dias ou até semanas. E é exatamente o que a Porto precisa: um sistema volátil, moderno e seguro.

### 1. READ ME PYTHON

O código entregue na 4ª sprint possui algumas particularidades como o uso da API VIACEP. Essa API busca o endereço do usuário com base em seu CEP, tendo em vista a integração do banco de dados interdisciplinar o endereço resposta dessa API não é salvo no banco pois em nossa solução não vimos necessidade de salvar o endereço do usuário. Tal API foi usada somente para efeitos demonstrativos de como seria o projeto final caso fosse adotado pela Porto Seguro. Outra particularidade é a necessidade de adaptação no arquivo functions.py, o avaliador deverá trocar o caminho na **linha 8 e na linha 15** para os caminhos de onde se encontra o login.json com suas informações e onde se encontra a pasta instaclient\_21\_11. Aqui segue os comandos de criação das tabelas utilizadas no código:

create table bicicleta ( cd\_bicicleta NUMERIC (4)constraint pk\_bicicleta primary key, id\_cliente NUMERIC (5), cd\_marca NUMERIC(4), cd\_foto NUMERIC (4), ds\_cor VARCHAR2(20), ds\_numero\_serie NUMERIC(9) NOT NULL UNIQUE, ds\_preco\_nota\_fiscal NUMBER (10,2), an\_ano\_compra date

);

CREATE TABLE cliente( id\_cliente NUMERIC (5) constraint pk\_cliente primary key, nm\_nome VARCHAR2 (60), numero\_cpf CHAR (11), numero\_rg CHAR (9), contato\_cliente VARCHAR2(15), dt\_nascimento DATE, el\_email VARCHAR2(100)

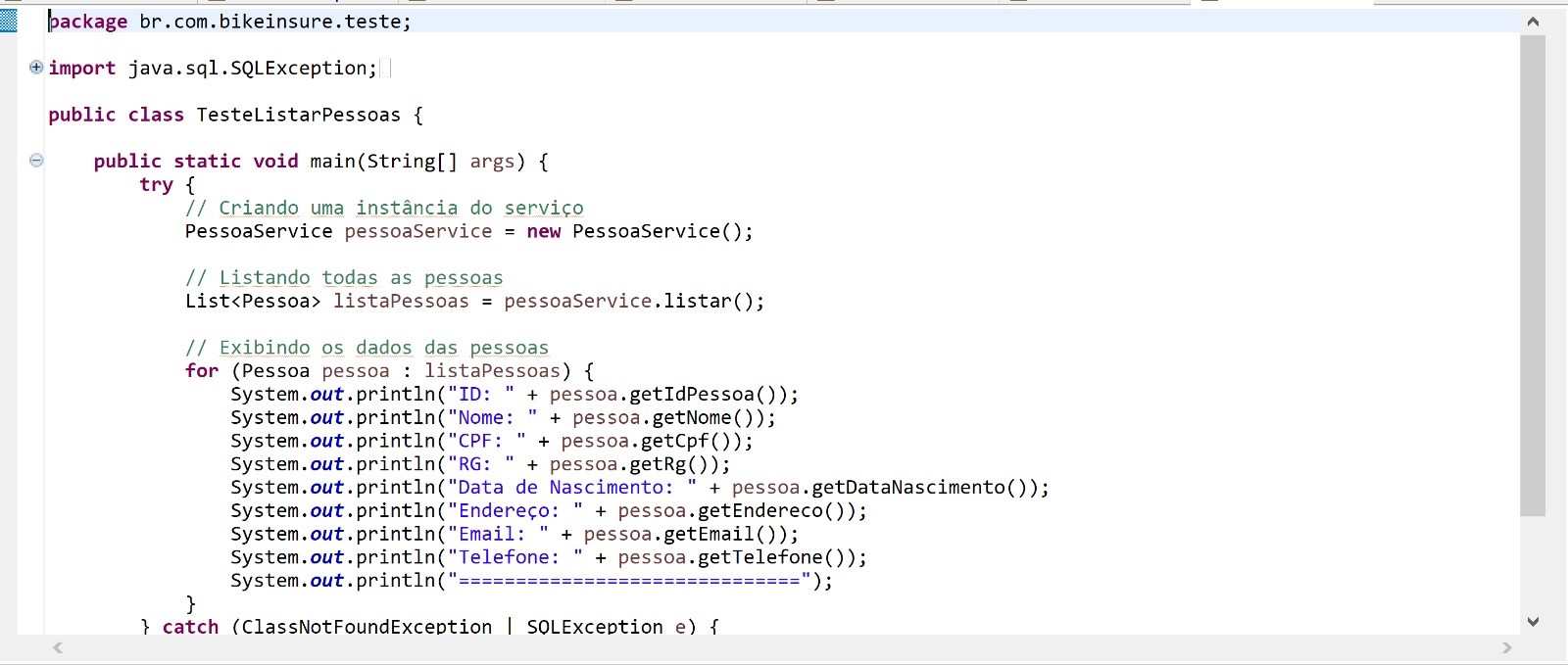
);

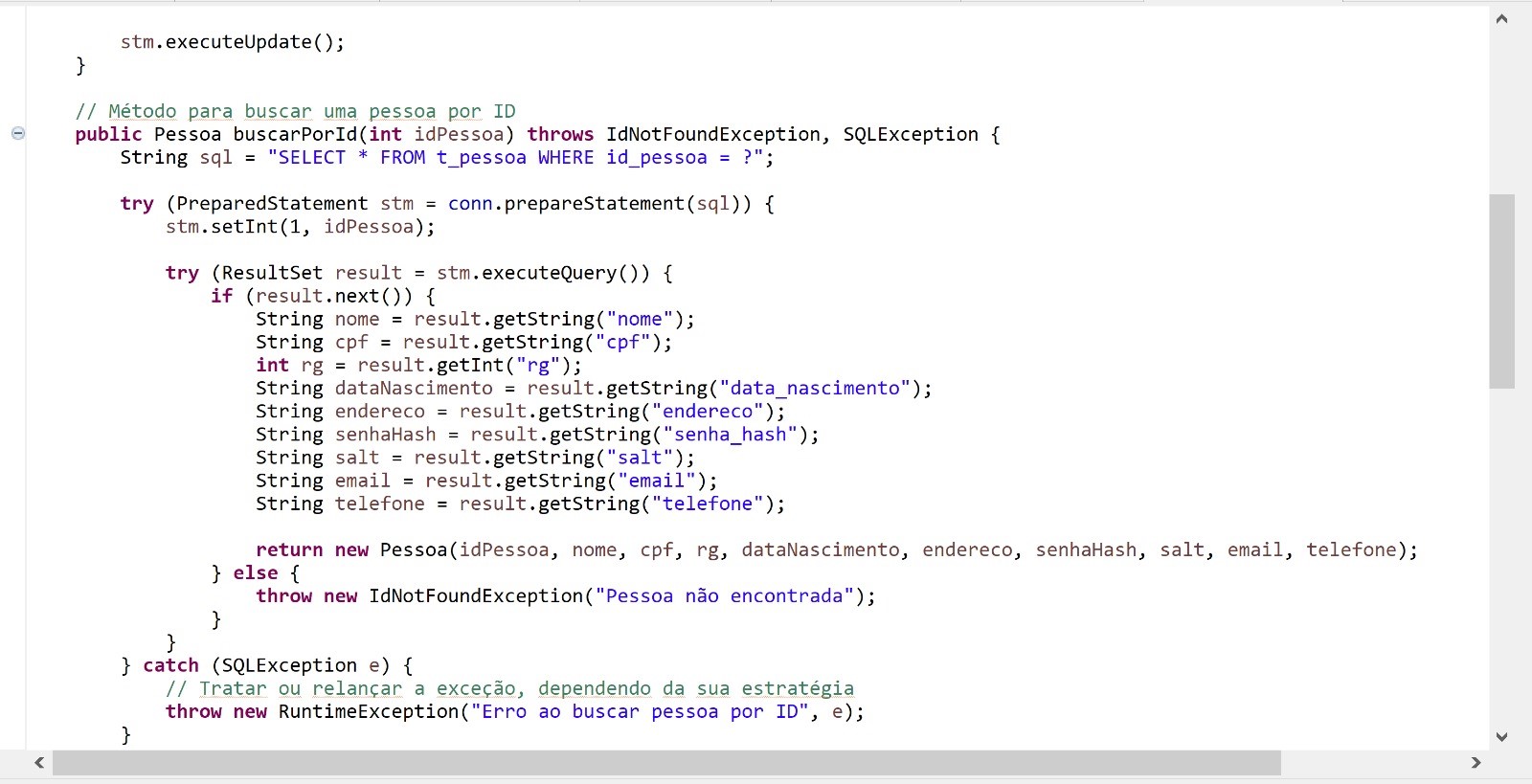
**2. LINK DO VÍDEO EXPLICATIVO**

[Pitch](https://www.youtube.com/watch?v=UXw2VwCRTx4) [entrega](https://www.youtube.com/watch?v=UXw2VwCRTx4) [Sprint](https://www.youtube.com/watch?v=UXw2VwCRTx4) [4](https://www.youtube.com/watch?v=UXw2VwCRTx4) [Python](https://www.youtube.com/watch?v=UXw2VwCRTx4) [2023](https://www.youtube.com/watch?v=UXw2VwCRTx4)

### 3. PRINTS DE TELAS COM LÓGICA DE PROGRAMAÇÃO

JAVA:

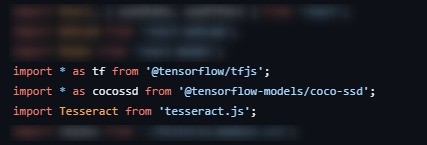
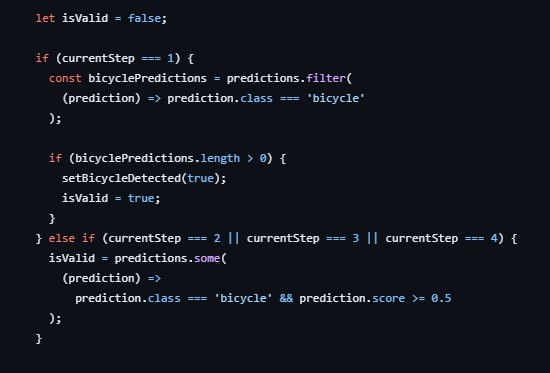






JS:





## Api Endpoints e HTTP Status Codes

| **Classe** | **URI** | **HTTP Verb** | **Response Status Code** |
| --- | --- | --- | --- |
| Bike | /api/bikes | GET | 200 OK |
| Bike | /api/bikes/{id} | GET | 200 OK ou 404 Not Found |
| Bike | /api/bikes/cadastrar | POST | 201 Created ou 500 Internal Server Error |
| Bike | /api/bikes/{id} | PUT | 200 OK ou 500 Internal Server Error |

| **Bike** | **/api/bikes/{id}** | **DELETE** | **200 OK ou 500 Internal Server Error** |
| --- | --- | --- | --- |
| Pessoa | /api/pessoas | GET | 200 OK |
| Pessoa | /api/pessoas/{id} | GET | 200 OK ou 404 Not Found |
| Pessoa | /api/pessoas/cadastrar | POST | 201 Created ou 500 Internal Server Error |
| Pessoa | /api/pessoas/{id} | PUT | 200 OK ou 500 Internal Server Error |

| **Pessoa** | **/api/pessoas/{id}** | **DELETE** | **200 OK ou 500 Internal Server Error** |
| --- | --- | --- | --- |
| Colaborador | /api/colaboradores | GET | 200 OK |
| Colaborador | /api/colaboradores/{id} | GET | 200 OK ou 404 Not Found |
| Colaborador | /api/colaboradores/cadastrar | POST | 201 Created ou 500 Internal Server Error |
| Colaborador | /api/colaboradores/{id} | PUT | 200 OK ou 500 Internal Server Error |

| **Colaborador** | **/api/colaboradores/{id}** | **DELETE** | **200 OK ou 500 Internal Server Error** |
| --- | --- | --- | --- |
| Seguro | /api/seguros | GET | 200 OK |
| Seguro | /api/seguros/{id} | GET | 200 OK ou 404 Not Found |
| Seguro | /api/seguros/cadastrar | POST | 201 Created ou 500 Internal Server Error |
| Seguro | /api/seguros/{id} | PUT | 200 OK ou 500 Internal Server Error |
| Seguro | /api/seguros/{id} | DELETE | 200 OK ou 500 Internal Server Error |