

UNIVERSIDADE ESTÁCIO DE SÁ
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
UNIDADE WEST SHOPPING – CAMPO GRANDE

**TRABALHO DE CADASTRO DE VEÍCULOS
EM DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES
EM PYTHON**

202203439634 – Victor Hugo Rodrigues Moreira de Souza

202202671011 – Wendell Velasco Soares

202202703397 – Daniel Filipe Vieira da Silva

202203617508 – Fabio Yudi Abe

202202136621 – Marcelo dos Santos e Santos

**Trabalho de Cadastro de Veículos
em Desenvolvimento Rápido de Aplicações em Python**

Trabalho de Cadastro de Veículos
apresentado a Universidade Estácio de Sá,
como exigência para avaliação na
disciplina Desenvolvimento Rápido de
Aplicações em Python

Orientador:

Prof. Ronaldo Candido dos Santos

SUMÁRIO

| | | |
|-----|-----------------------|---|
| 1 | INTRODUÇÃO | 3 |
| 1.1 | DESCRIÇÃO DO PROBLEMA | 3 |
| 1.2 | OBJETIVOS | 3 |
| 2 | DESENVOLVIMENTO | 4 |
| 2.1 | METODOLOGIA | 4 |
| 2.2 | PROPRIEDADES | 5 |
| 2.3 | EXECUÇÃO | 6 |
| 3 | CONCLUSÃO | 8 |
| | REFERÊNCIAS | 9 |

1 INTRODUÇÃO

É ideal que atualmente se tornou necessário que o carro tenha uma “identidade”, com suas identificações, características e o seu proprietário. Podendo ajudar na identificação de um veículo roubado e o dono deste carro, rastreando os veículos de uma forma mais eficiente. Sabendo até as peculiaridades de um veículo para uma identificação ainda mais precisa. Além de facilitar na revenda com o comprador podendo ver o que carro tem ou deixa de ter, suas características etc.

1.1 DESCRIÇÃO DO PROBLEMA

A falta de um cadastro pode prejudicar de diversas formas, como a falha na tentativa de recuperar o próprio carro em casos de roubo ou furto. E a falta de informações sobre o veículo trás outros problemas futuros, como a dificuldade para revenda, pois é muito mais difícil vender um carro sem as características e os detalhes documentados.

1.2 OBJETIVOS

Facilitar a vida das pessoas que tem seus carros e necessitam mostrar uma identificação imediata que aquele determinado carro e realmente desse indivíduo. Poder revender o carro com mais facilidade, mostrando os seus atributos e suas peculiaridades.

2 DESENVOLVIMENTO

O cadastro de veículos é essencial para o controle, gestão e segurança da frota automotiva em variados contextos, tanto públicos quanto privados. Este processo permite a obtenção de informações detalhadas sobre cada veículo, abrangendo características técnicas, histórico de manutenção e dados de propriedade e uso. Fundamentado em conceitos de gestão de frota, sistemas de informação geográfica (SIG) e tecnologias de identificação e rastreamento, o cadastro visa otimizar a operação da frota, garantir a segurança pública e privada, além de auxiliar na regulação e fiscalização. Contudo, desafios como a integração de sistemas, garantia da privacidade de dados e atualização contínua do cadastro permanecem como pontos em aberto a serem enfrentados. O contínuo aprimoramento do cadastro de veículos é fundamental para atender às demandas e acompanhar as tendências da mobilidade contemporânea.

2.1 METODOLOGIA

Para o desenvolvimento do nosso projeto, utilizamos a IDLE do Python, que é um ambiente de desenvolvimento integrado (IDE) bastante útil para a escrita, execução e depuração de código Python. A escolha do Python como linguagem de programação se deve à sua simplicidade, versatilidade e ampla gama de bibliotecas que facilitam o desenvolvimento de aplicações robustas.

Uma dessas bibliotecas é o Tkinter, que utilizamos para criar a interface gráfica do usuário (GUI). O Tkinter é uma das bibliotecas mais populares para a criação de GUIs em Python, pois oferece uma ampla variedade de widgets, como botões, rótulos, caixas de entrada, entre outros, permitindo que desenvolvedores construam interfaces de usuário de forma rápida e eficiente. Com o Tkinter, conseguimos criar uma interface intuitiva e amigável, que facilita a interação dos usuários com o nosso sistema.

Para o armazenamento e gerenciamento de dados, utilizamos a biblioteca SQLite, que é uma biblioteca de banco de dados SQL integrada. O SQLite é uma escolha excelente para projetos que necessitam de um banco de dados leve, rápido e sem a complexidade de um servidor de banco de dados completo. Ele armazena os dados em um único arquivo de banco de dados, o que simplifica a instalação e a manutenção do sistema. Além disso, o SQLite suporta a maioria dos comandos SQL, permitindo operações complexas de consulta e manipulação de dados.

Ao integrar o Tkinter com o SQLite, conseguimos desenvolver uma aplicação completa, com uma interface gráfica eficiente e um sistema de armazenamento de dados confiável. Este setup nos permitiu criar uma solução que atende às necessidades dos usuários de

maneira eficaz, garantindo uma experiência de uso satisfatória e um desempenho robusto do sistema.

2.2 PROPRIEDADES

Cadastro de carro

Número de chassi:

Nome do proprietário

CPF do proprietário

Placa

UF:

Cidade:

Marca:

Ano de fabricação:

Cor:

Quantidade de portas:

☒ 2 portas ☒ 4 portas

Possui mala:

☒ Sim ☒ Não

Capacidade da mala(L):

O carro possui:

☐ Ar-condicionado

☐ Radio

☐ Airbag

Problemas no carro:

Na fase de cadastro de veículos do nosso projeto, desenvolvemos uma interface gráfica intuitiva utilizando a biblioteca Tkinter. Esta interface foi cuidadosamente projetada para permitir a inserção de dados de forma eficiente e organizada, garantindo que todos os detalhes necessários sobre os veículos sejam capturados de maneira completa e precisa.

A interface de cadastro de veículos inclui diversos campos para o recebimento de dados. Cada campo foi criado com a finalidade de obter informações específicas que são essenciais para a gestão dos veículos cadastrados.

```

        cursor.execute('INSERT INTO Tabela1 VALUES (nome_prop, cpf_prop, placa, uf,
            (nome_prop, cpf_prop, placa, uf, cidade, marca, ano, cor, por

    connection.commit()
    print('Dados inseridos')
    connection.close()

def pegavalores():
    #SELECT
    rows = cursor.execute("SELECT * FROM Tabela1").fetchall()
    print(rows)

def Cadastro():
    root = tkinter.Tk()
    root.title("Cadastro")
    root.resizable(False, False)

    #TITULO
    label = tkinter.Label(root, text="Cadastro de carro")
    label.pack()

    #Nome do proprietário
    label = tkinter.Label(root, text="Nome do proprietário")
    label.pack()

    #campo de preenchimento do Nome do proprietário
    textoEntrada = tkinter.StringVar()
    e1 = tkinter.Entry(root)
    e1.bind('<Key>', lambda x:textoEntrada.set(e1.get()+x.char))
    e1.pack()

    #CPF do proprietário
    label = tkinter.Label(root, text="CPF do proprietário")
    label.pack()

    #campo de preenchimento do CPF do proprietário
    textoEntrada = tkinter.StringVar()
    e2 = tkinter.Entry(root)

```

A função responsável pelo cadastro de veículos é um elemento central do nosso sistema, e seu corpo foi cuidadosamente desenvolvido para garantir que todos os dados necessários sejam coletados e armazenados corretamente.

2.3 EXECUÇÃO

The screenshot displays a graphical user interface for a car registration system. It features two overlapping windows on a blue desktop background.

The top window, titled "Cadastro de carro", contains the following fields:

- Número de chassi: 2563477
- Nome do proprietário: (empty)

The bottom window, titled "Listagem dos carros", displays a list of registered vehicles:

Chassi: 25634 - Nome do Proprietário: Victor - CPF: 12525252 - Placa: 25fd24 - UF: AM - Cidade: - Marca: Fiat - Ano: 1983 - Cor: Cinza - Portas: - Mala: Não - Capacid:
Chassi: 2563477 - Nome do Proprietário: Victor Gabirle - CPF: 12525252 - Placa: 25fd24 - UF: AM - Cidade: - Marca: Fiat - Ano: 1983 - Cor: Cinza - Portas: - Mala: Não

Below the list, there is a form for adding a new car with the following fields and options:

- Ano de fabricação: 1983 (dropdown)
- Cor: Cinza (dropdown)
- Quantidade de portas: ☒ 2 portas ☒ 4 portas
- Possui mala: ☒ Sim ☒ Não
- Capacidade da mala(L): 0 (text input)
- O carro possui: ☒ Ar-condicionado ☐ Radio ☐ Airbag
- Problemas no carro: (text input)

At the bottom of the form are five buttons: Confirmar cadastro, Excluir cadastro, Atualizar cadastro, Carros cadastrados, and Procurar pelo chassi.

Na fase de execução do nosso sistema de cadastro de veículos, o usuário interage com a interface gráfica inserindo os dados necessários nos respectivos campos de atuação. Essa etapa é crucial, pois é aqui que as informações sobre os veículos são coletadas e posteriormente armazenadas no banco de dados.

3 CONCLUSÃO

Conclusão Final do Relatório de Cadastro de Veículos

O projeto de cadastro de veículos desenvolvido em Python utilizando a biblioteca Tkinter foi concluído com sucesso. Abaixo estão os principais pontos abordados e as funcionalidades implementadas:

Objetivo do Projeto

O objetivo principal foi criar uma aplicação desktop para gerenciar o cadastro de veículos, permitindo ao usuário inserir, visualizar e armazenar informações relevantes de forma intuitiva e eficiente.

Funcionalidades Implementadas

Interface Gráfica Intuitiva:

Janelas e Widgets: A interface gráfica foi criada utilizando Tkinter, com janelas bem organizadas e widgets apropriados, incluindo labels, entry boxes, comboboxes e botões.

Layout Responsivo: O layout foi desenvolvido para ser responsivo e fácil de navegar, garantindo uma experiência de usuário satisfatória.

Campos de Cadastro:

Informações Básicas: Campos para entrada de dados: Número de chassi, nome do proprietário, CPF do proprietário, paca, UF, cidade, marca, ano de fabricação, cor.

Validação de Dados: Implementação de validações para garantir que todos os campos obrigatórios sejam preenchidos corretamente antes de salvar. (É a parte final do texto na qual se apresentam as considerações finais. É a recapitulação sintética dos dados obtidos.

REFERÊNCIAS

1. Python.org.br - Este é o site oficial da comunidade Python no Brasil, que contém uma vasta gama de recursos, tutoriais e exemplos de código:
- <https://python.org.br> Acesso em: 06 Jun. 2024.
2. Blog da Alura - Alura é uma plataforma de educação que oferece muitos artigos sobre desenvolvimento de software, incluindo Python e Tkinter:
- <https://www.alura.com.br/artigos> Acesso em: 05 Jun. 2024.
3. DevMedia - Um portal popular para desenvolvedores, com tutoriais e artigos sobre várias linguagens de programação, incluindo Python:
- <https://www.devmedia.com.br> Acesso em: 04 Jun. 2024.
4. Curso em Vídeo - Canal de ensino de programação com cursos gratuitos, incluindo um curso de Python que abrange o uso do Tkinter:
- <https://www.cursoemvideo.com> Acesso em: 06 Jun. 2024.
5. Blog da Caelum* - Caelum é outra plataforma de educação em tecnologia que oferece artigos e tutoriais detalhados sobre desenvolvimento de software:
- <https://www.caelum.com.br> Acesso em: 05 Jun. 2024.
6. Python Café - Blog com foco em Python, que publica artigos e tutoriais práticos sobre várias bibliotecas e frameworks, incluindo Tkinter:
- <https://pythoncafe.com.br> Acesso em: 06 Jun. 2024.