Projeto integrador 3 – 11/05/2022

Administração em sistemas em nuvem

Nome: Giovanni Pereira Martins

Email: [giovanni.martins@accenture.com](mailto:giovanni.martins@accenture.com)

Título da Prática: Python Manipulando MySQL

Objetivos: Conectar o Python ao Banco de Dados MySQL e estabelecer a consulta deste BD em MySQL.

Ferramentas: Interpretador Python, Bando de Dados em MySQL

Atividade prática do projeto integrador: Embora tudo no universo da informática seja feito com base em dados: os sistemas e o que estes sistemas utilizam para funcionar, fato é que são os bancos de dados a verdadeira estrela do show. Um banco de dados deve ser visto como algo muito maior do que uma coleção de dados dispostos em linhas e colunas, pois aplicando o algoritmo correto, é capaz de revelar informações extremamente valiosas. Existe banco de dados para tudo, e sites como o IBGE apresentam bancos de dados sobre a população de uma região, dados climáticos, e muito mais, e ao se aplicar algum algoritmo podem oferecer informações que ajudam o Poder Público a criar política públicas que se destinam a melhorar a vida das pessoas. Portanto ter a expertise de lidar com banco de dados é vital a muitas profissões, principalmente as ligadas a tecnologia e ao desenvolvimento de sistemas, e como estamos na vibe do Python, que tal usá-lo para se conectar ao MySQL e criar um banco de dados?

**Passo 1.** **Antes de começar tenha certeza de que seu computador tem disponível um servidor MySQL como o “MySQL Community Server” e a biblioteca “MySQL Connector” e “Padas” para Python.**

*R- Instalado o Python versão 3.10.4 e o MySQL versão 8.0.28 no sistema operacional Windows 10 Enterprise 64 bit.*

**Passo 2. Comece com o Python e crie o código de acesso ao MySQL Community Server com conexão, host, user e password.**

*Foi criado um código em Python para conexão, verificação e informação caso haja algum erro durante a conexão. Foi utilizada estruturas condicionais para verificação de erro (“if, elif, else, try/except”). Na tela é exibida a informação de conexão estabelecida e do tipo de erro caso não haja a conexão.*

*Foram importadas as seguintes funções: “import mysql.connector” e “from mysql.connector import Error”. O código foi aproveitado da aplicação do exercício da pratica integradora anterior (10/05/2022).*

**

*Código em Python da conexão com banco de dados MySQL*

**Passo 3. Após ter sucesso com a conexão, crie seu banco de dados (create\_database), crie uma tabela simples, com ao menos 5 variáveis. Se desejar utilizar uma tabela existente, faça a devida citação e referência (conforme ABNT).**

*Foi criado um banco de dados com nome de “bd\_pratica3” diretamente no programa Mysql Workbench com 1 (uma) tabela de nome “tbl\_pratica3” contendo 05 colunas nas quais irão armazenar dados de veículos. As colunas receberam os nomes de “id”(número de identificação na tabela), “modelo”(modelo de um veículo), “marca”(marca do veículo), “ano”(ano de fabricação do veículo) e “placa”(placa do veículo).*

*As colunas foram criadas com as seguintes configurações:*

*- Id: Chave primaria e auto incrementada.*

*- modelo: Do tipo “VARCHAR” de 30 caracteres, não nula.*

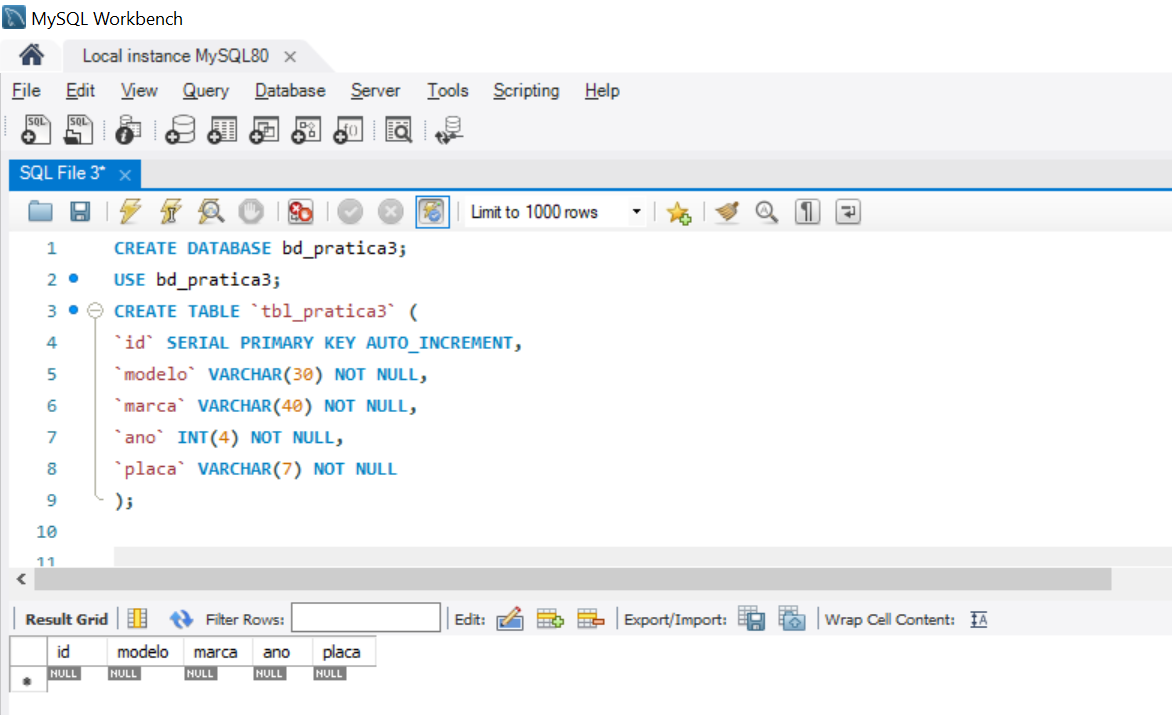
*- marca: Do tipo “VARCHAR” de 40 caracteres, não nula.*

*- ano: Do tipo “INT” de 4 (quatro) números, não nula.*

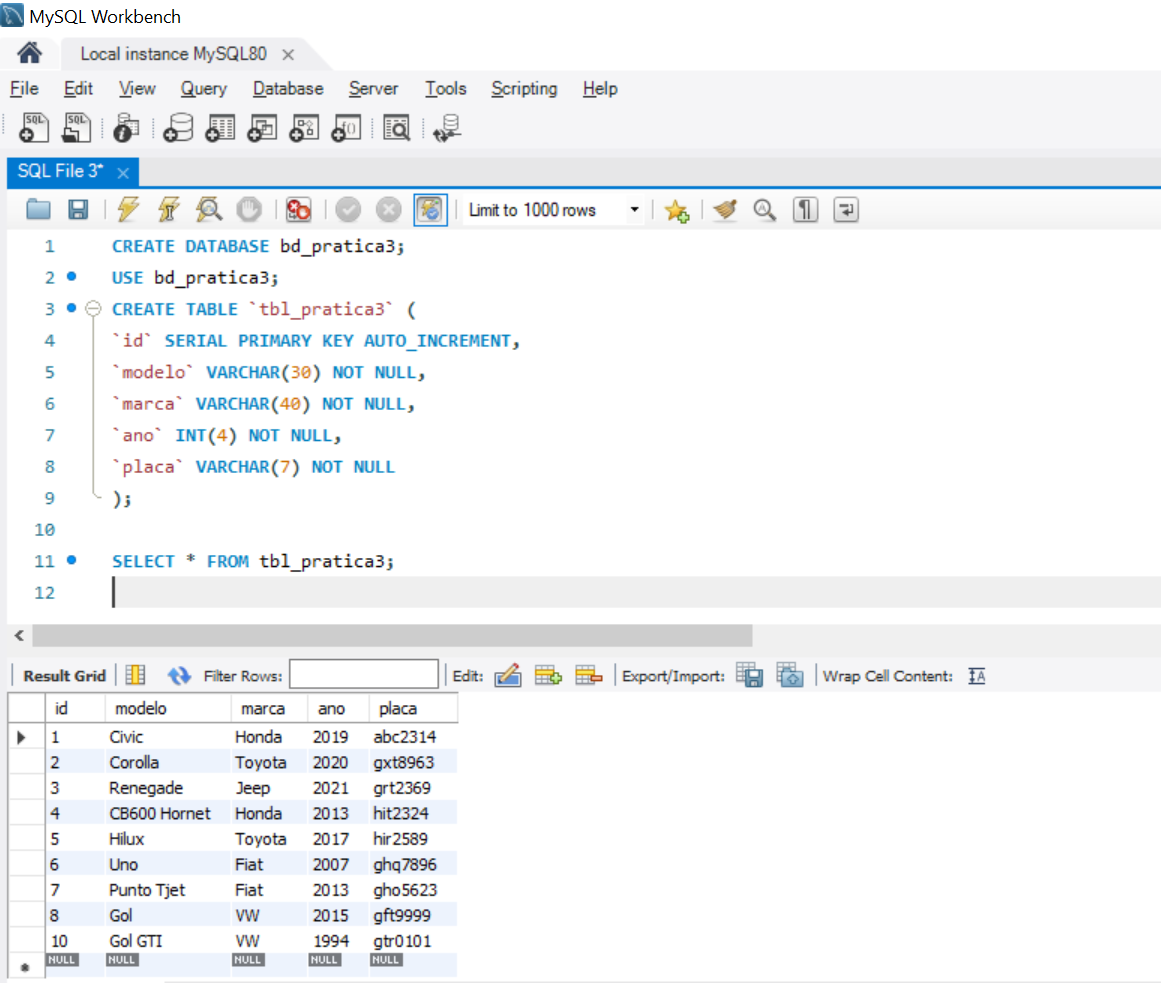
*- placa: Do tipo “VARCHAR” de 7 caracteres, não nula.*

**Passo 4. Assim que a tabela estiver pronta, mesmo sem dados, execute e faça o print da tela onde está a tabela sendo exibida.**

*Banco de dados criado:*



*Banco de dados preenchido via código python:*

**

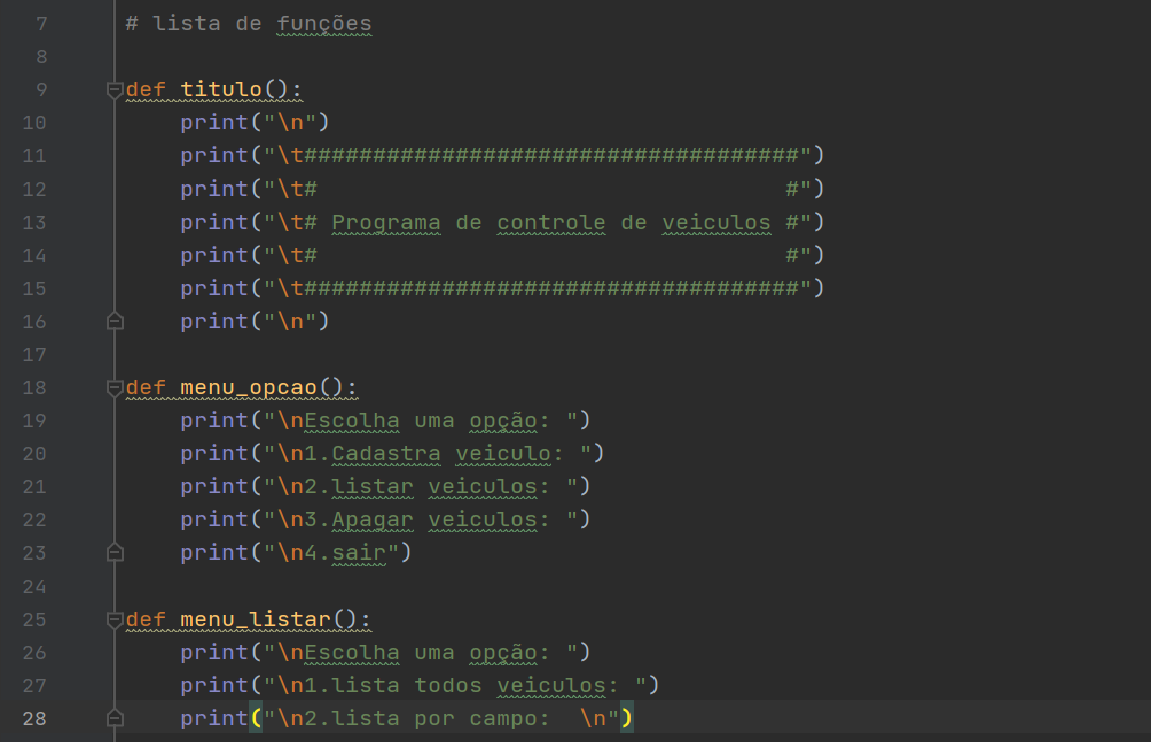
**Passo 5. Agora, com segunda parte de seu Dossiê, você deve salvar a imagem de seu progresso sendo no mínimo um “print” da tela com seu banco de dados sendo exibido pelo Python, apresentando um contexto para a imagem, ou seja, descreva do que se trata a imagem antes de inserir no documento de texto. Após a imagem, acrescente o código que resultou do processo de conexão do Python com o MySQL.**

*Para facilitar a criação e o entendimento, o código foi criado em 3 (três) partes. Conexão com banco de dados, criação das funções e estruturas condicionais com execução das funções. O código criado em Python realiza o acesso, inclusão, consulta e exclusão dos dados no banco de dados MySQL.*

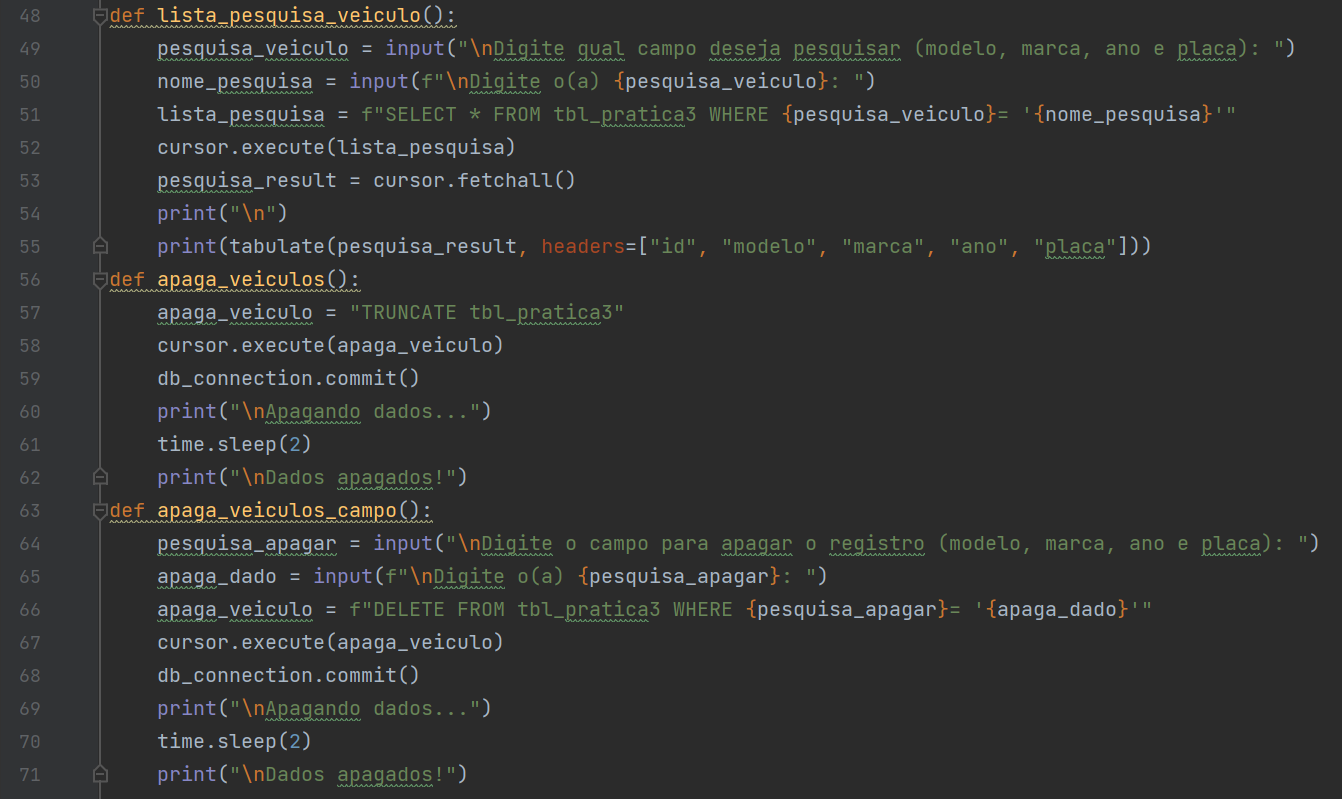
*O programa consiste em um controle de veículos onde os dados são cadastrados em um banco de dados e as informações podem ser manipuladas (Inclusão, consulta e exclusão de dados).*

*Para criação do código foram importados alguns módulos do Python: “import os”, “import time” e “import tabulate”.*

*O código foi criado com os principais comandos executados por funções. Foi uma forma de facilitar o processo de criação do código e fixar melhor o conhecimento adquirido nas aulas. Foram criadas 05 funções para titulo, menus, inclusão, consulta e exclusão no banco de dados.*

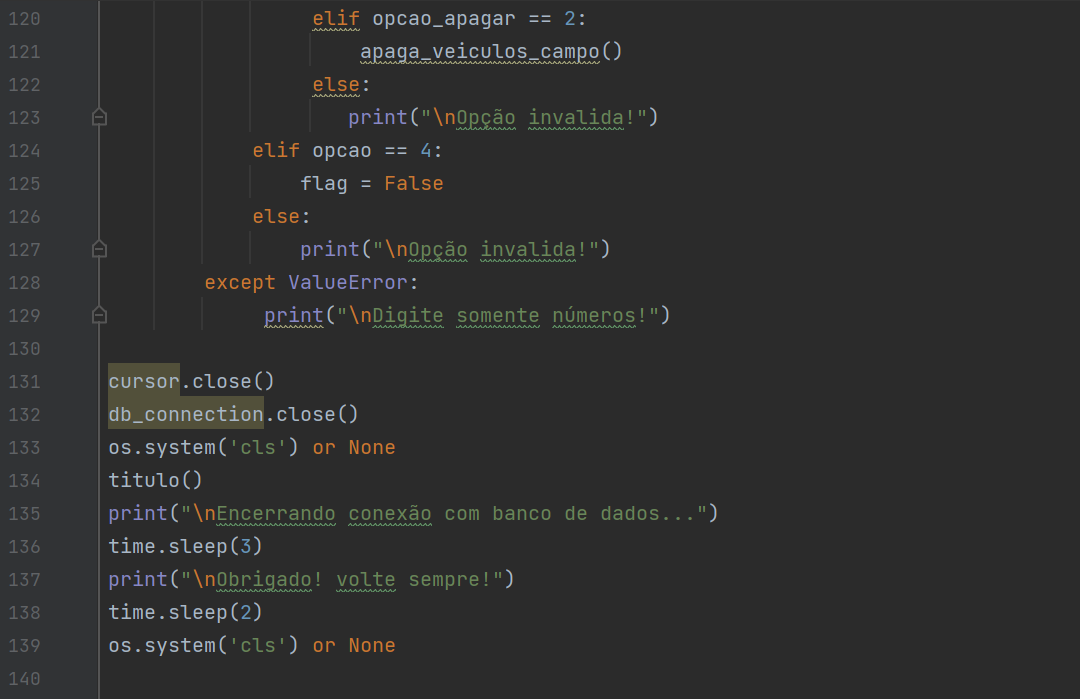
**

**

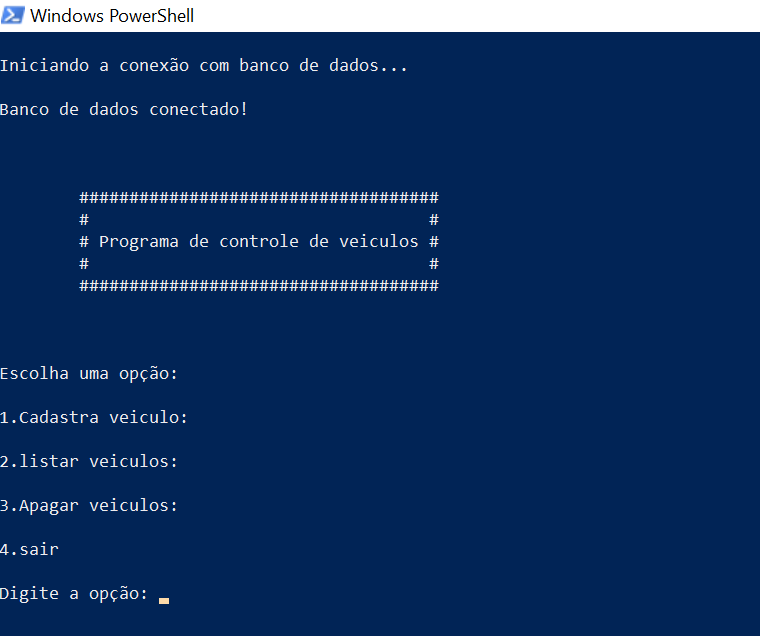
**

*Para chamar as funções e realizar tratativas de erros, foi criada no código estruturas de repetição e condicionais.*

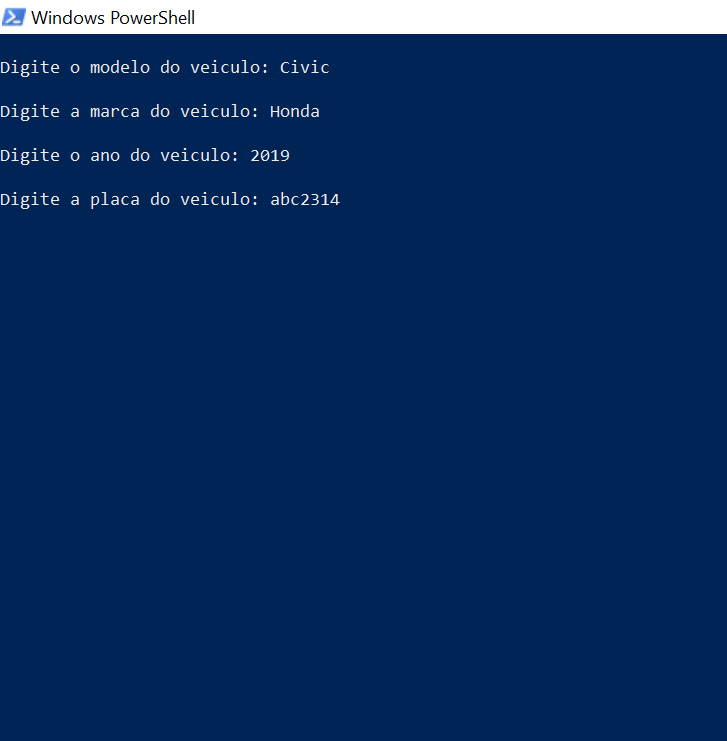
**

**

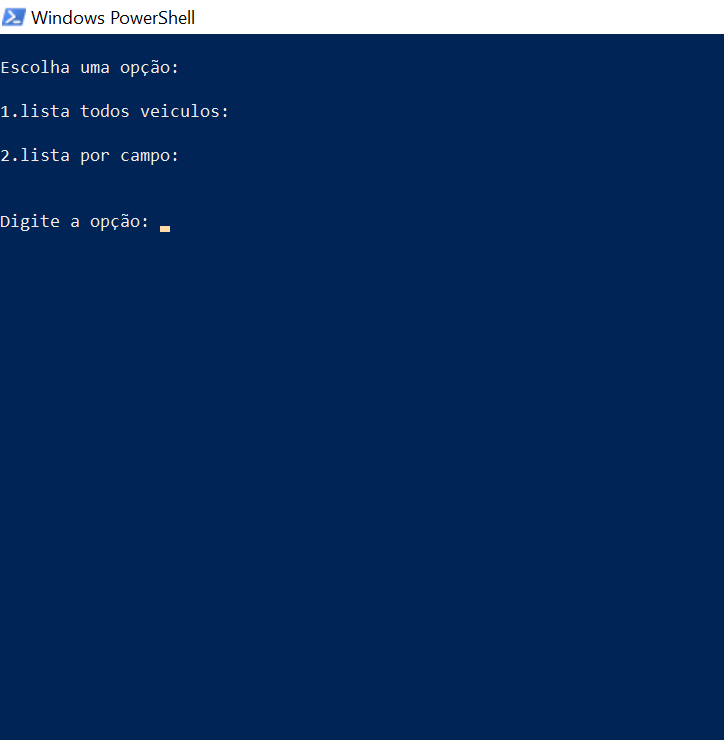
*Execução do programa:*

**

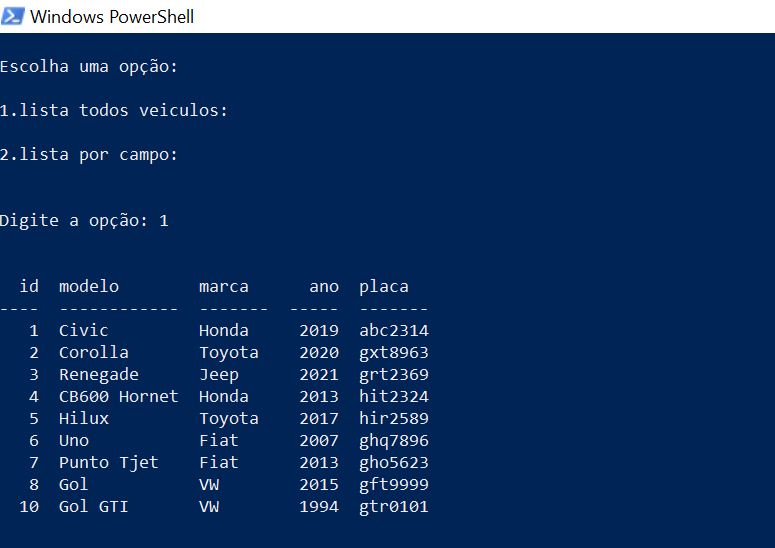
*Inclusão de veículos:*

**

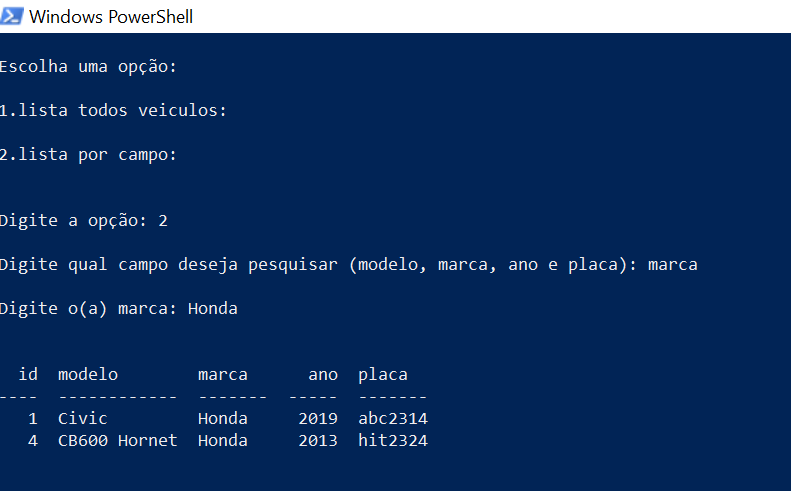
*Consulta da veículos:*

**

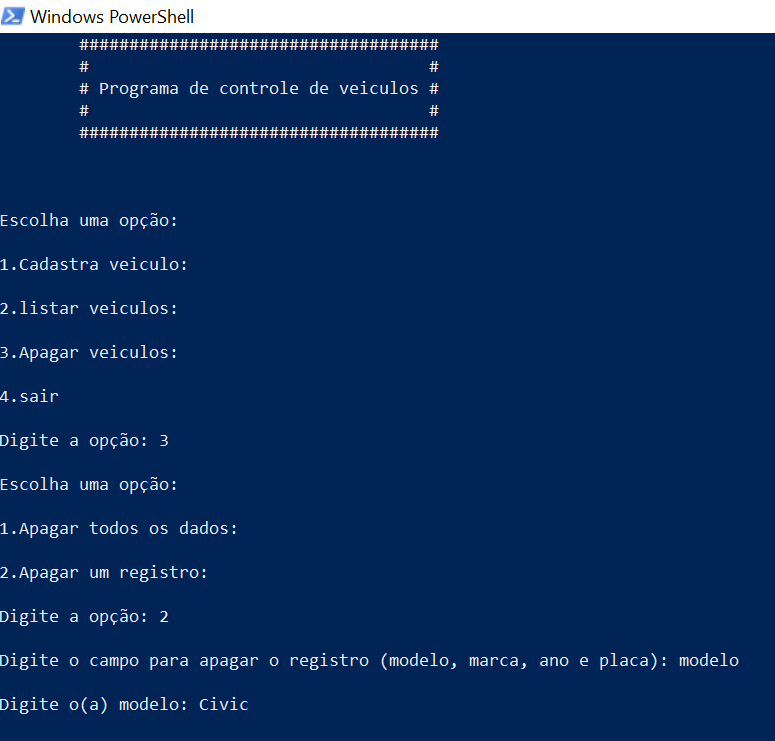
*Consulta toda tabela:*

**

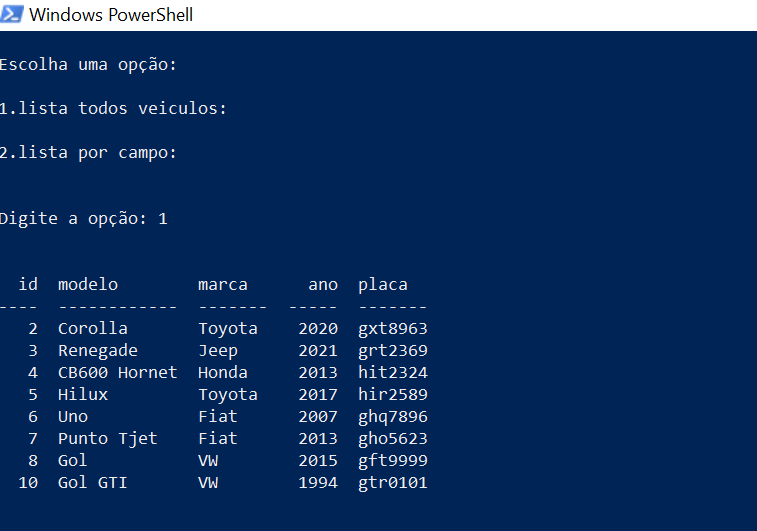
*Consulta por campo (coluna):*

**

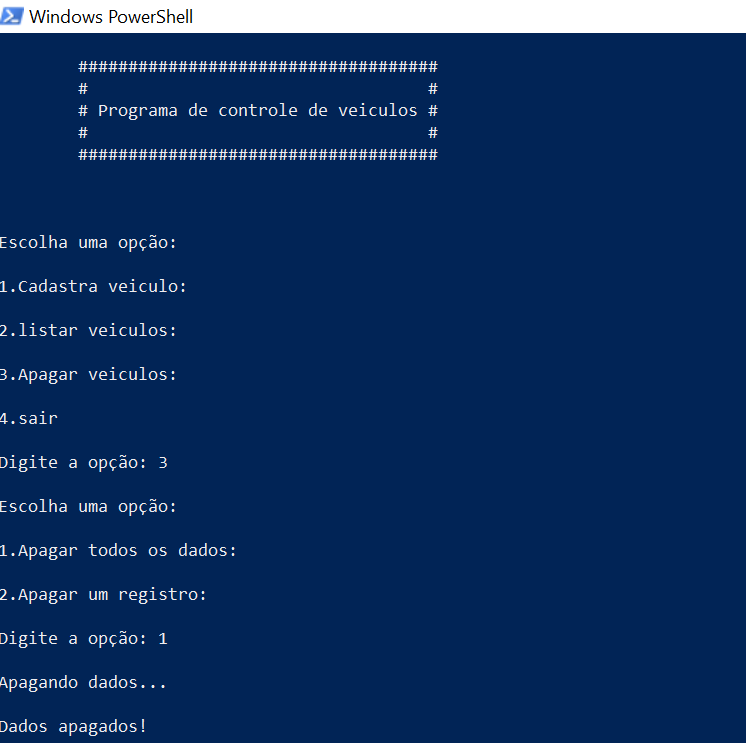
*Excluindo um registro:*

**

*Confirmando a exclusão do dado (Civic):*

**

*Apagando todos os dados:*

**

*Confirmando a exclusão dos registros:*

**

**Passo 6. Juntamente com as imagens contextualizadas, você deve responder as seguintes perguntas, em forma de texto corrido:**

1. **Agora que tem a habilidade de usar o MySQL com o Python, quais bancos de dados gostaria de criar e por quê?**

**R –** *Gostaria de criar de um banco de dados para criação de softwares de controle e análise de dados para empresas. Com base na estratificação desses dados, as empresas podem direcionar seus esforços e estratégias de negócios.*

1. **Em sua opinião, o que agrega a possibilidade de se ter acesso ao MySQL pelo Python?**

**R –** *O acesso ao banco de dados pelo Python eleva o nível de criação de aplicações. Podemos criar aplicações cada vez mais complexas e com uma maior agilidade. Podemos criar aplicações para acesso a bancos de dados já existentes, e automatizar cada vez serviços rotineiros.*

1. **Quais outros recursos, linguagens ou aplicações você julga que devem ser acessadas pelo Python?**

*R – Acredito que poderia ser acessado pelo Python, na área de automação industrial, como acesso a supervisórios de controle de máquinas e fazendo interação com linguagens mais antigas. A programação de aplicativos para telefonia móvel também seria uma área rica para desenvolvimento Python.*

1. **Qual temática da primeira semana de estudos você mais utilizou nesta atividade?**

*R – Utilizei a temática da disciplina de algoritmos com programação em Python. Mas principalmente, utilizei a temática das primeiras aulas de coaching e planejamento de carreira, onde foi abordado como ser mais produtivo. Com base nas aulas consegui nesta pratica utilizar o sendo de “importância x urgência onde consegui me organizar e realizas minha priorizações, conseguindo terminar o exercício em tempo hábil de com maior qualidade.*

1. **Faça um breve relato de sua experiência, e ao final compartilhe conosco o seguinte: você ensinaria alguém a fazer esta atividade?**

*R – A prática de hoje fez com que eu pudesse fixar meus conhecimentos adquiridos nas aulas de bancos de dados e programação. Conseguir realizar a prática proposta e implementar mais funcionalidades ao meu código. Sanei as minhas dúvidas com relação alguns comandos e e funcionalidades experimentando na prática. Ensinaria sim. Conforme meu dossiê criaria um passo a passo via vídeo, pois acredito ser mais lúdico e de fácil compressão.*

*Código completo:*

*import mysql.connector*

*from mysql.connector import Error*

*import os*

*import time*

*from tabulate import tabulate*

*# lista de funções*

*def titulo():*

*print("\n")*

*print("\t####################################")*

*print("\t# #")*

*print("\t# Programa de controle de veiculos #")*

*print("\t# #")*

*print("\t####################################")*

*print("\n")*

*def menu\_opcao():*

*print("\nEscolha uma opção: ")*

*print("\n1.Cadastra veiculo: ")*

*print("\n2.listar veiculos: ")*

*print("\n3.Apagar veiculos: ")*

*print("\n4.sair")*

*def menu\_listar():*

*print("\nEscolha uma opção: ")*

*print("\n1.lista todos veiculos: ")*

*print("\n2.lista por campo: \n")*

*def menu\_apagar():*

*print("\nEscolha uma opção: ")*

*print("\n1.Apagar todos os dados: ")*

*print("\n2.Apagar um registro: ")*

*def cadastra\_veiculo():*

*inf\_modelo = input("\nDigite o modelo do veiculo: ")*

*inf\_marca = input("\nDigite a marca do veiculo: ")*

*inf\_ano = input("\nDigite o ano do veiculo: ")*

*inf\_placa = input("\nDigite a placa do veiculo: ")*

*menu = (f'{inf\_modelo}', f'{inf\_marca}', inf\_ano, f'{inf\_placa}', )*

*sql = "INSERT INTO tbl\_pratica3 (modelo, marca, ano, placa) VALUES (%s, %s, %s, %s)"*

*cursor.execute(sql, menu)*

*db\_connection.commit()*

*cadastro = cursor.lastrowid*

*print("Foi cadastrado o novo usuario de id: ", cadastro)*

*def lista\_veiculos():*

*lista\_veiculo = "SELECT \* FROM tbl\_pratica3"*

*cursor.execute(lista\_veiculo)*

*veiculo\_result = cursor.fetchall()*

*print("\n")*

*print(tabulate(veiculo\_result, headers=["id", "modelo", "marca", "ano", "placa"]))*

*def lista\_pesquisa\_veiculo():*

*pesquisa\_veiculo = input("\nDigite qual campo deseja pesquisar (modelo, marca, ano e placa): ")*

*nome\_pesquisa = input(f"\nDigite o(a) {pesquisa\_veiculo}: ")*

*lista\_pesquisa = f"SELECT \* FROM tbl\_pratica3 WHERE {pesquisa\_veiculo}= '{nome\_pesquisa}'"*

*cursor.execute(lista\_pesquisa)*

*pesquisa\_result = cursor.fetchall()*

*print("\n")*

*print(tabulate(pesquisa\_result, headers=["id", "modelo", "marca", "ano", "placa"]))*

*def apaga\_veiculos():*

*apaga\_veiculo = "TRUNCATE tbl\_pratica3"*

*cursor.execute(apaga\_veiculo)*

*db\_connection.commit()*

*print("\nApagando dados...")*

*time.sleep(2)*

*print("\nDados apagados!")*

*def apaga\_veiculos\_campo():*

*pesquisa\_apagar = input("\nDigite o campo para apagar o registro (modelo, marca, ano e placa): ")*

*apaga\_dado = input(f"\nDigite o(a) {pesquisa\_apagar}: ")*

*apaga\_veiculo = f"DELETE FROM tbl\_pratica3 WHERE {pesquisa\_apagar}= '{apaga\_dado}'"*

*cursor.execute(apaga\_veiculo)*

*db\_connection.commit()*

*print("\nApagando dados...")*

*time.sleep(2)*

*print("\nDados apagados!")*

*# Conexão com banco de dados*

*os.system('cls') or None*

*print("\nIniciando a conexão com banco de dados...")*

*time.sleep(3)*

*try:*

*db\_connection = mysql.connector.connect(host='localhost', user='root', password='Master02\*',*

*database='bd\_pratica3')*

*print("\nBanco de dados conectado!\n")*

*time.sleep(3)*

*except mysql.connector.Error as error:*

*if error.errno == errorcode.ER\_BAD\_DB\_ERROR:*

*print("\nBanco de dados não existe!")*

*elif error.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:*

*print("\nLogin ou senha incorretos!")*

*else:*

*print(error)*

*else:*

*cursor = db\_connection.cursor()*

*# Instruções condicionais*

*flag = True*

*while flag:*

*titulo()*

*try:*

*menu\_opcao()*

*opcao = int(input("\nDigite a opção: "))*

*if opcao == 1:*

*os.system('cls') or None*

*cadastra\_veiculo()*

*elif opcao == 2:*

*os.system('cls') or None*

*menu\_listar()*

*opcao\_lista = int(input("\nDigite a opção: "))*

*if opcao\_lista == 1:*

*lista\_veiculos()*

*elif opcao\_lista == 2:*

*lista\_pesquisa\_veiculo()*

*else:*

*print("\nOpção invalida!")*

*elif opcao == 3:*

*menu\_apagar()*

*opcao\_apagar = int(input("\nDigite a opção: "))*

*if opcao\_apagar == 1:*

*apaga\_veiculos()*

*elif opcao\_apagar == 2:*

*apaga\_veiculos\_campo()*

*else:*

*print("\nOpção invalida!")*

*elif opcao == 4:*

*flag = False*

*else:*

*print("\nOpção invalida!")*

*except ValueError:*

*print("\nDigite somente números!")*

*cursor.close()*

*db\_connection.close()*

*os.system('cls') or None*

*titulo()*

*print("\nEncerrando conexão com banco de dados...")*

*time.sleep(3)*

*print("\nObrigado! volte sempre!")*

*time.sleep(2)*

*os.system('cls') or None*