Nome:

QUESTÕES

#Carregamento de bibliotecas   
import numpy as np   
import matplotlib.pyplot as plt   
import pandas as pd   
   
#Carregamento de arquivo / Entrada de dados   
arquivo = "C:\\Users\\dsadm\\Desktop\\Aula\_17\_04\\arquivo\_histograma.csv" #nome do arquivo   
dados = pd.read\_csv(arquivo,header=0, delimiter=';') #le o arquivo de dados   
dados\_dict = dados.to\_dict("list") #converte p/ o formato dicionario   
   
#Tratando os dados   
   
# calculo do numero de bins para o histograma   
# Formula de Sturges   
N = len(dados\_dict["Imposto"]) #acha o numero de imposto   
k = 1+3.32\*np.log10(N) #calcula o numero de BINS   
k = np.round(k) #arredonda o valor achado de BINS   
   
#Apresentação de resultados   
   
print(dados\_dict)   
print("O número de BINS para esse histograma é: ", k)   
#Desenha o histograma   
plt.hist(dados\_dict["Imposto"], bins=int(k)) #Faz o histograma e converte para numero inteiro o numero de BINS   
plt.title("Histograma dos Impostos")   
plt.xlabel("Taxa do Imposto")   
plt.ylabel("Quantidade")   
plt.show()

[AUT\_PROJ\_INT.project](https://sesisenaispedu-my.sharepoint.com/personal/leonardo_oliveira167_senaisp_edu_br/Documents/Arquivos%20de%20Chat%20do%20Microsoft%20Teams/AUT_PROJ_INT.project)

import mysql.connector  
  
# Estabelecer a conexão com o banco de dados  
conexao = mysql.connector.connect(  
 host='localhost',  
 user='root',  
 password='',  
)  
  
print(conexao)

auth\_plugin='mysql\_native\_password'

import PySimpleGUI as sg  
  
  
layout = [  
 [sg.Text('Usuário')],  
  
 [sg.Input(key='usuario', do\_not\_clear=False)],  
  
 [sg.Text('Senha')],  
  
 [sg.Input(key='senha', password\_char='\*',  
 do\_not\_clear=False)],  
  
 [sg.Button('Cadastrar')],  
  
 [sg.Button('Login')],  
  
 [sg.Text('', key='mensagem')],  
]  
  
window = sg.Window('Login', layout=layout)  
  
layoutCalculadora = [  
 [sg.Text("Calculadora Senai")],  
 [sg.Text("Multiplicação: ")],  
 [sg.Text("Soma: ")],  
 [sg.Text("Subtração: ")],  
 [sg.Text("Divisão: ")],  
]  
  
Usuario = "Leonardo"  
Senha = "Thiago"  
  
while True:  
 event, values = window.read()  
 if event == sg.WINDOW\_CLOSED:  
  
 break  
 elif event == 'Cadastrar':  
  
 if (len(values['usuario'].strip()) == 6) and (len(values['senha'].strip()) == 6):  
 usuario\_criado = values['usuario'].lower()  
  
 senha\_criada = values['senha']  
  
 window['mensagem'].update('Usuário ,"' +  
 usuario\_criado + '" criado com sucesso!')  
  
 else:  
 window['mensagem'].update('Preenchimento Inválido!')  
  
 elif event == 'Login':  
  
 usuario = values['usuario']  
 senha = values['senha']  
  
 if senha == Senha and usuario == Usuario:  
  
 window['mensagem'].update('Login realizado com sucesso!')  
  
 else:  
 window['mensagem'].update('Usuario ou senha incorretos!')

# Layout da calculadora  
layout\_calculadora = [  
 [sg.Text("Calculadora Senai")],  
 [sg.Text("Multiplicação: ")],  
 [sg.Input(key='multiplicar\_1', do\_not\_clear=False)],  
 [sg.Input(key='multiplicar\_2', do\_not\_clear=False)],  
 [sg.Button('Multiplicar')],  
 [sg.Text("Soma: ")],  
 [sg.Input(key='somar\_1', do\_not\_clear=False)],  
 [sg.Input(key='somar\_2', do\_not\_clear=False)],  
 [sg.Button('Somar')],  
 [sg.Text("Subtração: ")],  
 [sg.Input(key='subtrair\_1', do\_not\_clear=False)],  
 [sg.Input(key='subtrair\_2', do\_not\_clear=False)],  
 [sg.Button('Subtrair')],  
 [sg.Text("Divisão: ")],  
 [sg.Input(key='dividir\_1', do\_not\_clear=False)],  
 [sg.Input(key='dividir\_2', do\_not\_clear=False)],  
 [sg.Button('Dividir')],  
 [sg.Text('', key='resultado')],  
]

import PySimpleGUI as sg  
  
layout = [  
 [sg.Text('Usuário')],  
  
 [sg.Input(key='usuario', do\_not\_clear=False)],  
  
 [sg.Text('Senha')],  
  
 [sg.Input(key='senha', password\_char='\*',  
 do\_not\_clear=False)],  
  
 [sg.Button('Cadastrar')],  
  
 [sg.Button('Login')],  
  
 [sg.Text('', key='mensagem')],  
]  
  
window = sg.Window('Login', layout=layout)  
  
layoutCalculadora = [  
 [sg.Text("Calculadora Senai")],  
 [sg.Text("Resultado:")],  
 [sg.Text('', key="Resultado")],  
  
 [sg.Text("Multiplicação: ")],  
 [sg.Input(key='multiplicar\_1', do\_not\_clear=False)],  
 [sg.Input(key='multiplicar\_2', do\_not\_clear=False)],  
 [sg.Button('Multiplicar', key='Mult', )],  
 [sg.Text("Soma: ")],  
 [sg.Input(key='somar\_1', do\_not\_clear=False)],  
 [sg.Input(key='somar\_2', do\_not\_clear=False)],  
 [sg.Button('Somar', key='Soma')],  
 [sg.Text("Subtração: ")],  
 [sg.Input(key='subtrair\_1', do\_not\_clear=False)],  
 [sg.Input(key='subtrair\_2', do\_not\_clear=False)],  
 [sg.Button('Subtrair', key='Sub', )],  
 [sg.Text("Divisão: ")],  
 [sg.Input(key='dividir\_1', do\_not\_clear=False)],  
 [sg.Input(key='dividir\_2', do\_not\_clear=False)],  
 [sg.Button('Dividir', key='Div', )],  
 [sg.Button('Sair', key='Sair', )],  
  
]  
  
  
windowCal = sg.Window('Calculadora', layout=layoutCalculadora)  
Usuario = "Leonardo"  
Senha = "Thiago"  
  
while True:  
 event, values = window.read()  
 if event == sg.WINDOW\_CLOSED:  
  
 break  
 elif event == 'Cadastrar':  
  
 if (len(values['usuario'].strip()) == 6) and (len(values['senha'].strip()) == 6):  
 usuario\_criado = values['usuario'].lower()  
  
 senha\_criada = values['senha']  
  
 window['mensagem'].update('Usuário ,"' +  
 usuario\_criado + '" criado com sucesso!')  
  
 else:  
 window['mensagem'].update('Preenchimento Inválido!')  
  
 elif event == 'Login':  
  
 usuario = values['usuario']  
 senha = values['senha']  
  
 if senha == Senha and usuario == Usuario:  
  
 window['mensagem'].update('Login realizado com sucesso!')  
 window.close()  
  
 while True:  
  
 eventCal, valuesCal = windowCal.read()  
  
 if eventCal == "Mult":  
 Val1 = valuesCal["multiplicar\_1"]  
 Val2 = valuesCal["multiplicar\_2"]  
 if Val1 == '' or Val2 == "":  
 windowCal['Resultado'].update("Insira todos os campos necessários!!")  
 else:  
  
 resultado = float(Val1) \* float(Val2)  
  
 windowCal['Resultado'].update(str(resultado))  
 elif eventCal == "Soma":  
 Val1 = valuesCal["somar\_1"]  
 Val2 = valuesCal["somar\_2"]  
 if Val1 == '' or Val2 == "":  
 windowCal['Resultado'].update("Insira todos os campos necessários!!")  
 else:  
 resultado = float(Val1) + float(Val2)  
  
 windowCal['Resultado'].update(str(resultado))  
 elif eventCal == "Sub":  
 Val1 = valuesCal["subtrair\_1"]  
 Val2 = valuesCal["subtrair\_2"]  
 if Val1 == '' or Val2 == "":  
 windowCal['Resultado'].update("Insira todos os campos necessários!!")  
 else:  
 resultado = float(Val1) - float(Val2)  
  
 windowCal['Resultado'].update(str(resultado))  
 elif eventCal == "Div":  
 Val1 = valuesCal["dividir\_1"]  
 Val2 = valuesCal["dividir\_2"]  
 if Val1 == '' or Val2 == "":  
 windowCal['Resultado'].update("Insira todos os campos necessários!!")  
 else:  
  
 try:  
 resultado = float(Val1) / float(Val2)  
 windowCal['Resultado'].update(str(float(resultado)))  
 except ZeroDivisionError:  
 windowCal['Resultado'].update("Não pode dividir zero por zero!!")  
  
  
 elif eventCal == "Sair":  
 break  
 else:  
 window['mensagem'].update('Usuario ou senha incorretos!')

import pymongo  
import time  
import os  
from pyModbusTCP.client import ModbusClient  
  
# myclient = pymongo.MongoClient("mongodb+srv://thiagocontato1232:tNBa2wr7XO6EODbm@cluster0.ob27bb5.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0")  
  
# mydb = myclient["Teste"]  
# mycol = mydb['PecasGrandes']  
  
# x = mycol.insert\_one({"Peca":"peca\_peq"})  
# exit()  
c = ModbusClient(host="192.168.160.110", port=502, unit\_id=1, auto\_open=True)  
  
while True:  
 try:  
 regsHold = c.read\_holding\_registers(0, 3)  
  
 except Exception as erro:  
 print(erro.args)  
 else:  
 if regsHold:  
  
 peca\_peq = regsHold[0]  
 peca\_med = regsHold[1]  
 peca\_grande = regsHold[2]  
 print("Contador peça Pequena", peca\_peq)  
 print("Contador peça Média", peca\_med)  
 print("Contador peça Grande", peca\_grande)  
 print("Leitura bem sucedida!")  
  
 else:  
 print("read error")  
  
 time.sleep(5)  
 os.system('cls')

[Aula\_21\_05](https://sesisenaispedu-my.sharepoint.com/:f:/g/personal/leonardo_oliveira167_senaisp_edu_br/EhBZqPMTnOtOl6o5Xl1MAz8B9W7OzzBBoxJa9xMe5OPgFw?email=thiago.alves33%40senaisp.edu.br&e=OCl9Hd)

Pasta com arquivos da somativa de automação. Diagrama eletropneumatico e uma parte do ladder

git config --global --unset-all

[Eng de Software Ativ03\_William e Leonardo.docx](https://sesisenaispedu-my.sharepoint.com/personal/william_costa11_senaisp_edu_br/Documents/Arquivos%20de%20Chat%20do%20Microsoft%20Teams/Eng%20de%20Software%20Ativ03_William%20e%20Leonardo.docx)

[Eng de Software Ativ03\_William e Leonardo.docx](https://sesisenaispedu-my.sharepoint.com/personal/leonardo_oliveira167_senaisp_edu_br/Documents/Arquivos%20de%20Chat%20do%20Microsoft%20Teams/Eng%20de%20Software%20Ativ03_William%20e%20Leonardo.docx)

[library-backend.zip](https://sesisenaispedu-my.sharepoint.com/personal/leonardo_oliveira167_senaisp_edu_br/Documents/Arquivos%20de%20Chat%20do%20Microsoft%20Teams/library-backend.zip)

Professor,

Nós da turma B de ADS, Thiago, Leonardo, Isadora e William, estamos com uma ideia de aprender e poder criar portfolio em análise de dados. Gostaríamos de saber se o Sr. poderia nos ajudar a encontrar empresas, pessoas ou projetos reais para executarmos gratuitamente, apenas pelo aprendizado.

Pensamos que pela sua experiência na área, poderia nos auxiliar a dar os primeiros passos.

Thiago Henrique Benedito Alves - Questões BackEnd

1. **Identificar as partes principais de um CRUD em Node.js e Express.**

Identifique e explique cada uma das seguintes partes:

• Configuração do servidor.

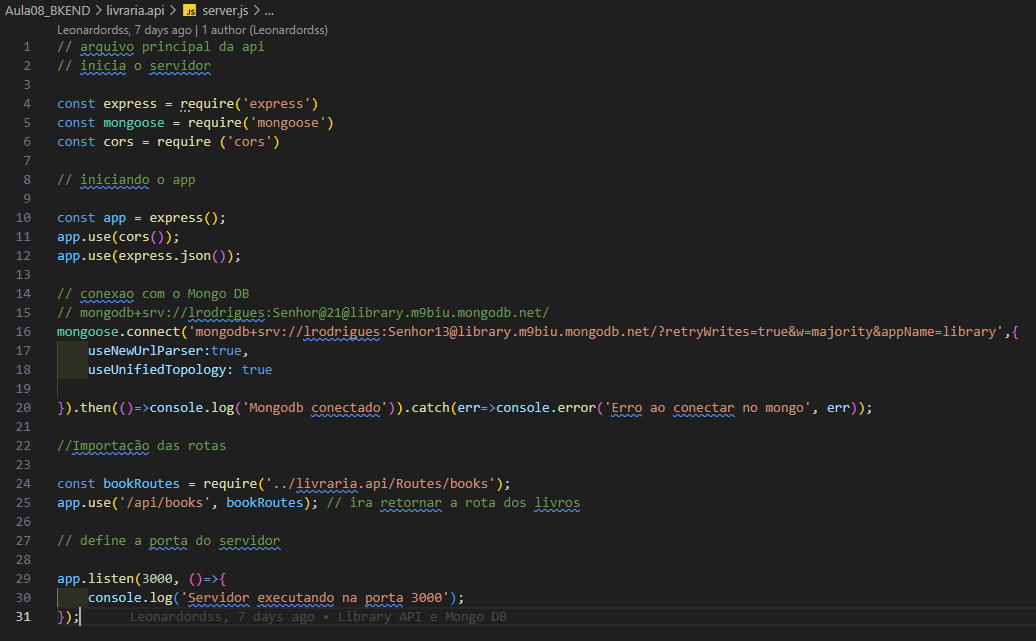
• Rota para listar os itens.

• Rota para adicionar um item.

• Conexão com o banco de dados.

**RESPOSTA:**

Esta parte do código é responsável por inicializar o servidor Express e configurar middlewares, como o uso de CORS e o parser de JSON.



**Explicação:**

Express: O Express é utilizado para criar o servidor e definir rotas.

Mongoose: Usado para conectar ao MongoDB, o banco de dados no qual os dados serão armazenados.

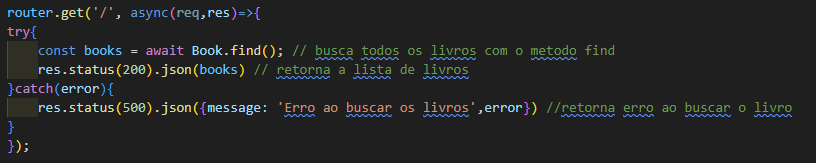
CORS: Permite que o servidor aceite requisições de diferentes origens, o que é útil em aplicações web.

JSON Parser: O express.json() permite que o servidor interprete o corpo das requisições no formato JSON.

Porta do Servidor: O servidor escuta na porta 3000.

* **Rota para Listar os Itens (GET)**

Essa rota permite recuperar todos os livros do banco de dados.



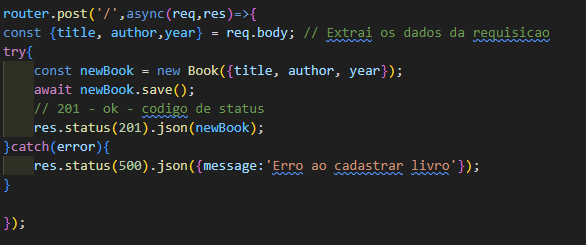
**Explicação:**

Método GET: Utilizado para buscar recursos.

Book.find(): Método do Mongoose que busca todos os documentos da coleção "books".

Resposta: Retorna um status 200 e a lista de livros em formato JSON.

* **Rota para Adicionar um Item (POST)**

Essa rota permite adicionar um novo livro ao banco de dados.

**Explicação:**

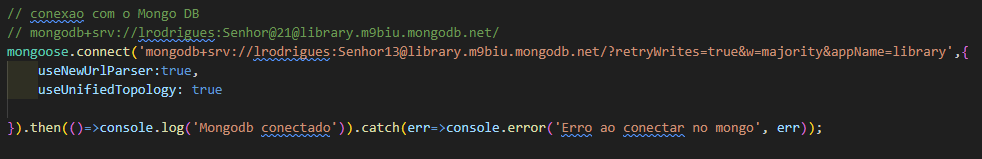
Método POST: Utilizado para criar novos recursos.

Criação do Livro: Um novo objeto Book é criado e salvo no banco de dados.

Resposta: Retorna um status 201, indicando que o recurso foi criado com sucesso, junto com os dados do novo livro.

* **Conexão com o Banco de Dados**

Esta parte do código configura a conexão com o MongoDB usando o Mongoose.



**Explicação:**

String de Conexão: Fornece as credenciais e o endereço do banco de dados.

useNewUrlParser e useUnifiedTopology: Opções de configuração do Mongoose para garantir que a conexão seja realizada corretamente.

Tratamento de Erros: O uso de .catch permite lidar com falhas na conexão, informando no console.

**2. O que é um servidor e qual sua função no back-end de uma aplicação?**

3. Qual a função do Node.js no desenvolvimento back-end?

4. Como funcionam as rotas no Express.js e como são definidas?

5. O que são controladores (controllers) no contexto do back-end e como eles organizam a lógica de negócio?

6. O que é uma requisição HTTP e quais são suas partes principais?

7. O que são middlewares e como eles podem ser aplicados em rotas específicas no Express?

8. Como o Mongoose é utilizado para MongoDB?

9.O que é validação de dados no back-end e como ela pode ser feita?

10.O que é validação de dados no back-end e como ela pode ser feita com Mongoose?

11.Como o Express.js é utilizado para criar rotas que permitem realizar as operações CRUD no banco de dados MongoDB?

12. Qual é a função do modelo Book.js (usando Mongoose) no sistema, e como ele define a estrutura de um documento no MongoDB?

13.Como a função findByldAndUpdate é utilizada na rota de atualização (PUT) para modificar os dados de um livro?

14.Por que utilizamos try...catch nas operações CRUD e como tratamos erros em requisições que falham?

15. Como o método find() no Mongoose funciona para retornar a lista de todos os livros na rota GET, e como ele interage com o MongoDB?

16. O que o comando npm init -y faz ao criar um novo projeto Node.js? 17. O que o método app.get() faz na API Express?

18. Qual a função da linha app.listen(port, () => {...}); no código da API?

19.Qual é o papel do require('express') no início do código?

20. Como você pode testar se a API está funcionando corretamente sem o usar o navegador?