Rafael Giordano Matesco 2ºH

Desafios arquivo texto

b)

def gerarHTML(reg=0):  
 try:  
 diretorio = 'R:/UNIVAP/2 ano/4 bimestre/'  
 extensao = '.html'  
 reg = str(reg)  
 arquivo = open(diretorio+reg+extensao, 'w')  
 except Exception as erro:  
 print(f'Ocorreu um erro: {erro}')  
 else:  
 try:  
 reg = int(reg)  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select \* from disciplinasxprofessores where codprofessor = {reg} order by curso;')  
 tabela = comandosql.fetchall()  
 comandosql.close()  
 body = ''  
 conteudo = ''  
 cursoAnterior = 0  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 discHTML = ''  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select nomeprof from professores where registro = {registro[2]};')  
 nomeProfessores = comandosql.fetchone()  
 for v in nomeProfessores:  
 nomeProf = v  
 comandosql.close()  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select curso from disciplinasxprofessores where curso = {registro[3]};')  
 cursos = comandosql.fetchone()  
 for v in cursos:  
 curso = v  
 comandosql.close()  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select codigodisc, nomedisc from disciplinas where codigodisc = {registro[1]};')  
 disciplina = comandosql.fetchall()  
 comandosql.close()  
  
 for v in disciplina:  
 discHTML += f'''  
 {v[0]} | {v[1]} <br>  
 '''  
 if cursoAnterior != curso:  
 body += f'''  
 <br><br>  
 Curso: {curso} <br>  
 CÓDIGO DA DISCIPLINA | NOME DISCIPLINA <br>  
 '''  
 body += f'''  
 {discHTML}  
 '''  
 cursoAnterior = curso  
 conteudo += f'''<html>  
 <head>  
 <title>Professor {nomeProf}</title>  
 </head>  
 <body>  
 Disciplinas do professor: {nomeProf} <br>  
 {body}  
 </body>  
 </html>  
 '''  
 arquivo.write(conteudo)  
  
 else:  
 print('Não existem cadastros!!!')  
 except Exception as erro:  
 print(f'Ocorreu erro: {erro}')  
 arquivo.close()

Desafios arquivo PDF

def gerarPDF(escolha1=0,escolha2=0):  
 try:  
 nomeArquivo = input('Digite o nome do arquivo PDF:')  
 pdf = canvas.Canvas(f'{nomeArquivo}.pdf')  
 regua(pdf)  
 pdf.setFillColor('black')  
 pdf.setTitle(nomeArquivo)  
  
 #professor  
 if escolha1 == 1:  
 #nome  
 if escolha2 == 1:  
 try:  
 nome = input('Digite o nome do professor: ')  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f"select \* from professores where nomeprof like '{nome}%';")  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 nomeProf = list()  
 telProf = list()  
 idadeProf = list()  
 salarioProf = list()  
 for indice, registro in enumerate(tabela):  
 nomeProf.append(registro[1])  
 telProf.append(registro[2])  
 idadeProf.append(registro[3])  
 salarioProf.append(registro[4])  
 except Exception as error:  
 print(f'Ocorreu erro: {error}')  
  
 try:  
 x = 780  
 for y in range(0,len(nomeProf)):  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf[y]}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf[y]}')  
 x -= 30  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 except:  
 x = 780  
 for y in range(0, qnt, 1):  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf[y]}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf[y]}')  
 x -= 20  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 #registro  
 elif escolha2 == 2:  
 try:  
 reg = int(input('Digite o registro do professor: '))  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {reg};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 nomeProf = registro[1]  
 telProf = registro[2]  
 idadeProf = registro[3]  
 salarioProf = registro[4]  
 except Exception as error:  
 print(f'Ocorreu erro: {error}')  
  
 try:  
 x = 780  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf}')  
  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 except:  
 x = 780  
 pdf.setFont('Helvetica-Oblique', 36)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf}')  
 pdf.setFont('Helvetica-Oblique', 16)  
 x -= 20  
 pdf.drawString(67, x, f'telefone -> {telProf}')  
 x -= 20  
 pdf.drawString(67, x, f'idade -> {idadeProf}')  
 x -= 20  
 pdf.drawString(67, x, f'salário -> {salarioProf}')  
  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 #curso  
 elif escolha1 == 2:  
 #disciplina  
 if escolha2 == 1:  
 print('')  
 #professor  
 elif escolha2 == 2:  
 print('')  
 #carga horaria  
 elif escolha2 == 3:  
 print('')  
 except Exception as error:  
 print(f'{error}')

CÓDIGO COMPLETO

from prettytable import PrettyTable  
import mysql.connector  
from reportlab.pdfgen import canvas  
  
def abrebanco():  
 try:  
 print()  
 global conexao  
 conexao = mysql.connector.Connect(host='localhost',database='univap',user='root', password='')  
 if conexao.is\_connected():  
 informacaobanco = conexao.get\_server\_info()  
 print(f'Conectado ao servidor banco de dados - Versão {informacaobanco}')  
 global comandosql  
 comandosql = conexao.cursor()  
 comandosql.execute('select database();')  
 nomebanco = comandosql.fetchone()  
 print(f'Banco de dados acessado = {nomebanco}')  
 print()  
 return 1  
  
 else:  
 print('Conexão não realizada com banco')  
 return 0  
 except Exception as erro:  
 print(f'Erro : {erro}')  
 return 0  
  
  
def mostratodasdisc():  
 grid = PrettyTable(['Códigos das Disciplinas', "Nomes de Disciplinas"])  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas;')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 grid.add\_row([registro[0], registro[1]])  
 print(grid)  
 print('\n')  
 else:  
 print('Não existem disciplinas cadastradas!!!')  
 except Exception as erro:  
 print(f'Ocorreu erro: {erro}')  
def consultardisciplina(cd=0):  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas where codigodisc = {cd};')  
  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 print(f'Nome da Disciplina: {registro[1]} ')  
 return 'cadastrado'  
 else:  
 return 'naocadastrado'  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta disciplina: Erro===>>> {error}')  
def cadastrardisciplina(cd=0,nd=''):  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'insert into disciplinas(codigodisc, nomedisc) values({cd},"{nd}") ;')  
 conexao.commit()  
 print('\n')  
 return 'Cadastro da disciplina realizado com sucesso !!!! '  
  
 except Exception as erro :  
 print('\n')  
 return 'não foi possivel cadastrar disciplina'  
def alterardisciplina(cd=0, nomedisciplina=''):  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas where codigodisc = {cd};')  
  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'Não existe essa disciplina. '  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta disciplina: Erro===>>> {error}')  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'Update disciplinas SET nomedisc="{nomedisciplina}" where codigodisc = {cd};')  
 conexao.commit()  
 print('\n')  
 return 'Disciplina alterada com sucesso !!! '  
 except Exception as erro :  
 print('\n')  
 return 'Não foi possível alterada esta disciplina'  
def excluirdisciplina(cd=0):  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas where codigodisc = {cd};')  
  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'Não existe essa disciplina. '  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta disciplina: Erro===>>> {error}')  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'delete from disciplinas where codigodisc = {cd};')  
 conexao.commit()  
 print('\n')  
 return 'Disciplina excluída com sucesso !!! '  
 except Exception as erro :  
 print('\n')  
 return 'Não foi possível excluir esta disciplina'  
  
def mostratodasprof():  
  
 grid = PrettyTable(['Registro', "Nome do professor", "Telefone", "Idade", "Salário"])  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores;')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 grid.add\_row([registro[0], registro[1], registro[2], registro[3], registro[4]])  
 print(grid)  
 print('\n')  
 else:  
 print('Não existem professores cadastrados!!!')  
 except Exception as erro:  
 print(f'Ocorreu erro: {erro}')  
def consultarprofessor(reg=0):  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {reg};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
  
 for registro in tabela:  
 print(f'Nome do professor: {registro[1]}')  
 print(f'telefone: {registro[2]}')  
 print(f'Idade: {registro[3]}')  
 print(f'Salário: {registro[4]}')  
 print('\n')  
 return 'cadastrado'  
 else:  
 return 'naocadastrado'  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar este professor')  
def cadastrarprofessor(reg=0,np='',tp='',ip=0, sp=0.0):  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where nomeprof = "{np}";')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 1:  
 return 'já existe professor com esse nome'  
 comandosql = conexao.cursor()  
 comandosql.execute(f'insert into professores(registro, nomeprof, telefoneprof, idadeprof, salarioprof) values({reg},"{np}","{tp}", {ip}, {sp});')  
 conexao.commit()  
 tabela=comandosql.fetchall()  
  
 print('\n')  
 return 'Cadastro do professor realizado com sucesso !!!! '  
 except Exception as erro :  
 print(erro)  
 return 'Não foi possível cadastrar este professor !!!'  
def alterarprofessor(reg=0, nomeprofessor='', telefoneprof='', idadeprof=0, salarioprof=0.0):  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {reg};')  
  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'Não existe esse professor. '  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta disciplina: Erro===>>> {error}')  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'Update professores SET nomeprof="{nomeprofessor}", telefoneprof="{telefoneprof}", idadeprof="{idadeprof}", salarioprof="{salarioprof}" where registro = {reg};')  
 conexao.commit()  
 print('\n')  
 return 'Professor alterado com sucesso !!! '  
 except Exception as erro :  
  
 return 'Não foi possível alterar este professor'  
def excluirprofessor(reg=0):  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {reg};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'Esse prof não existe.'  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar este professor')  
 try:  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'delete from professores where registro = {reg};')  
 conexao.commit()  
 print('\n')  
 return 'Professor excluído com sucesso !!! '  
 except Exception as erro :  
 print(erro)  
 return 'Não foi possível excluir este professor'  
  
def mostratodasdiscxprof():  
 print('\n')  
 grid = PrettyTable(['Código', "Cód. Disc.", "Cód. prof.", "Curso", "Carga horária", "Ano letivo"])  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinasxprofessores;')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 grid.add\_row([registro[0], registro[1], registro[2], registro[3], registro[4], registro[5]])  
 print(grid)  
 print('\n')  
 else:  
 print('Não existem cadastros!!!')  
 except Exception as erro:  
 print(f'Ocorreu erro')  
def consultardiscxprof(cod=0):  
 print('\n')  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinasxprofessores where codigodisciplinanocurso = {cod};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 print(f'código disciplina: {registro[1]}')  
 print(f'código professor: {registro[2]}')  
 print(f'curso: {registro[3]}')  
 print(f'carga horária: {registro[4]}')  
 print(f'ano letivo: {registro[5]}')  
 print('\n')  
 return 'cadastrado'  
 else:  
 return 'naocadastrado'  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta tabela')  
def cadastrardiscxprof(cod=0,coddisc=0,codprof=0,curso=0, cargaHoraria=0, anoLetivo=0):  
 print('\n')  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas where codigodisc = {coddisc};')  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'disciplina não existe'  
  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {codprof};')  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'professor não existe'  
  
 comandosql = conexao.cursor()  
 comandosql.execute(f'insert into disciplinasxprofessores(codigodisciplinanocurso, coddisciplina, codprofessor, curso,cargahoraria, anoletivo) values({cod},{coddisc}, {codprof}, {curso}, {cargaHoraria}, {anoLetivo});')  
 conexao.commit()  
 print('\n')  
 return 'Cadastro da tabela realizado com sucesso !!!! '  
 except Exception as erro :  
  
 return 'Não foi possível cadastrar este registro na tabela !!!'  
def alterardiscxprof(cod=0,coddisc=0,codprof=0,curso=0, cargaHoraria=0, anoLetivo=0):  
 print('\n')  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinas where codigodisc = {coddisc};')  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return'disciplina não existe'  
  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {codprof};')  
  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'professor não existe'  
  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinasxprofessores where codigodisciplinanocurso = {cod};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'registro não existe'  
  
 comandosql = conexao.cursor()  
 comandosql.execute(f'Update disciplinasxprofessores SET coddisciplina = {coddisc}, codprofessor = {codprof}, curso = {curso}, cargahoraria = {cargaHoraria}, anoletivo = {anoLetivo} where codigodisciplinanocurso = {cod};')  
  
  
 print('\n')  
 return 'Tabela alterada com sucesso !!! '  
  
 except Exception as erro :  
  
 return 'Não foi possível alterar esta tabela'  
def excluirdiscxprof(cod=0):  
 print('\n')  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from disciplinasxprofessores where codigodisciplinanocurso = {cod};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount == 0:  
 return 'Esse registro não existe.'  
 except Exception as error:  
 return (f'Ocorreu erro ao tentar consultar esta tabela')  
 try:  
 comandosql = conexao.cursor()  
 comandosql.execute(f'delete from disciplinasxprofessores where codigodisciplinanocurso = {cod};')  
 conexao.commit()  
 print('\n')  
 return 'registro excluído com sucesso !!! '  
 except Exception as erro :  
  
 return 'Não foi possível excluir este registro'  
  
def gerarHTML(reg=0):  
 try:  
 diretorio = 'R:/UNIVAP/2 ano/4 bimestre/'  
 extensao = '.html'  
 reg = str(reg)  
 arquivo = open(diretorio+reg+extensao, 'w')  
 except Exception as erro:  
 print(f'Ocorreu um erro: {erro}')  
 else:  
 try:  
 reg = int(reg)  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select \* from disciplinasxprofessores where codprofessor = {reg} order by curso;')  
 tabela = comandosql.fetchall()  
 comandosql.close()  
 body = ''  
 conteudo = ''  
 cursoAnterior = 0  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 discHTML = ''  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select nomeprof from professores where registro = {registro[2]};')  
 nomeProfessores = comandosql.fetchone()  
 for v in nomeProfessores:  
 nomeProf = v  
 comandosql.close()  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select curso from disciplinasxprofessores where curso = {registro[3]};')  
 cursos = comandosql.fetchone()  
 for v in cursos:  
 curso = v  
 comandosql.close()  
  
 comandosql = conexao.cursor(buffered=True)  
 comandosql.execute(f'select codigodisc, nomedisc from disciplinas where codigodisc = {registro[1]};')  
 disciplina = comandosql.fetchall()  
 comandosql.close()  
  
 for v in disciplina:  
 discHTML += f'''  
 {v[0]} | {v[1]} <br>  
 '''  
 if cursoAnterior != curso:  
 body += f'''  
 <br><br>  
 Curso: {curso} <br>  
 CÓDIGO DA DISCIPLINA | NOME DISCIPLINA <br>  
 '''  
 body += f'''  
 {discHTML}  
 '''  
 cursoAnterior = curso  
 conteudo += f'''<html>  
 <head>  
 <title>Professor {nomeProf}</title>  
 </head>  
 <body>  
 Disciplinas do professor: {nomeProf} <br>  
 {body}  
 </body>  
 </html>  
 '''  
 arquivo.write(conteudo)  
  
 else:  
 print('Não existem cadastros!!!')  
 except Exception as erro:  
 print(f'Ocorreu erro: {erro}')  
 arquivo.close()  
  
def regua(pdf):  
 pdf.setFillColor('red')  
 for coluna in range(0,595,5):  
 pdf.setFont('Helvetica-Oblique', 2)  
 pdf.drawString(coluna,0,f'{coluna}')  
 for linha in range(0,841,5):  
 pdf.setFont('Helvetica-Oblique', 2)  
 pdf.drawString(0,linha,f'{linha}')  
  
def gerarPDF(escolha1=0,escolha2=0):  
 try:  
 nomeArquivo = input('Digite o nome do arquivo PDF:')  
 pdf = canvas.Canvas(f'{nomeArquivo}.pdf')  
 regua(pdf)  
 pdf.setFillColor('black')  
 pdf.setTitle(nomeArquivo)  
  
 #professor  
 if escolha1 == 1:  
 #nome  
 if escolha2 == 1:  
 try:  
 nome = input('Digite o nome do professor: ')  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f"select \* from professores where nomeprof like '{nome}%';")  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 nomeProf = list()  
 telProf = list()  
 idadeProf = list()  
 salarioProf = list()  
 for indice, registro in enumerate(tabela):  
 nomeProf.append(registro[1])  
 telProf.append(registro[2])  
 idadeProf.append(registro[3])  
 salarioProf.append(registro[4])  
 except Exception as error:  
 print(f'Ocorreu erro: {error}')  
  
 try:  
 x = 780  
 for y in range(0,len(nomeProf)):  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf[y]}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf[y]}')  
 x -= 30  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 except:  
 x = 780  
 for y in range(0, qnt, 1):  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf[y]}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf[y]}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf[y]}')  
 x -= 20  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 #registro  
 elif escolha2 == 2:  
 try:  
 reg = int(input('Digite o registro do professor: '))  
 print('\n')  
 comandosql = conexao.cursor()  
 comandosql.execute(f'select \* from professores where registro = {reg};')  
 tabela = comandosql.fetchall()  
 if comandosql.rowcount > 0:  
 for registro in tabela:  
 nomeProf = registro[1]  
 telProf = registro[2]  
 idadeProf = registro[3]  
 salarioProf = registro[4]  
 except Exception as error:  
 print(f'Ocorreu erro: {error}')  
  
 try:  
 x = 780  
 pdf.setFont('Helvetica-Oblique', 24)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf}')  
 pdf.setFont('Helvetica-Oblique', 12)  
 x -= 20  
 pdf.drawString(137, x, f'telefone -> {telProf}')  
 x -= 20  
 pdf.drawString(137, x, f'idade -> {idadeProf}')  
 x -= 20  
 pdf.drawString(137, x, f'salário -> {salarioProf}')  
  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 except:  
 x = 780  
 pdf.setFont('Helvetica-Oblique', 36)  
 pdf.drawString(107, x, f'Dados do professor {nomeProf}')  
 pdf.setFont('Helvetica-Oblique', 16)  
 x -= 20  
 pdf.drawString(67, x, f'telefone -> {telProf}')  
 x -= 20  
 pdf.drawString(67, x, f'idade -> {idadeProf}')  
 x -= 20  
 pdf.drawString(67, x, f'salário -> {salarioProf}')  
  
 pdf.save()  
 print(f'{nomeArquivo} foi gerado com sucesso!')  
 #curso  
 elif escolha1 == 2:  
 #disciplina  
 if escolha2 == 1:  
 print('')  
 #professor  
 elif escolha2 == 2:  
 print('')  
 #carga horaria  
 elif escolha2 == 3:  
 print('')  
 except Exception as error:  
 print(f'{error}')  
  
  
  
  
  
  
  
#=========================================================================  
  
  
while True:  
 try:  
 resposta = int(input('Digite: 1-tabela disciplinas | 2- tabela professores | 3-tabela disciplinas X professores | 4-Gerar HTML do professor | 5-Gerar PDF do professor | 6-sair do programa: '))  
 except:  
 resposta = int(input('Digite: 1-tabela disciplinas | 2- tabela professores | 3-tabela disciplinas X professores | 4-sair do programa: '))  
 while resposta<1 or resposta>4:  
 resposta = int(input(  
 'Digite: 1-tabela disciplinas | 2- tabela professores | 3-tabela disciplinas X professores | 4-sair do programa: '))  
 if resposta == 1:  
 if abrebanco() == 1:  
 print('='\*80)  
 print('{:^80}'.format('SISTEMA UNIVAP - DISCIPLINAS'))  
 print('='\*80)  
 while True:  
 try:  
 resps = int(input('Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar uma disciplina \n 2-cadastrar uma disciplina \n 3-alterar o nome de uma disciplina \n 4-excluir uma disciplina \n 5-sair da tabela: '))  
 except:  
 resps = int(input('Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar uma disciplina \n 2-cadastrar uma disciplina \n 3-alterar o nome de uma disciplina \n 4-excluir uma disciplina \n 5-sair da tabela: '))  
  
 while (resps < 0) or (resps > 5):  
 resps = int(input('Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar uma disciplina \n 2-cadastrar uma disciplina \n 3-alterar o nome de uma disciplina \n 4-excluir uma disciplina \n 5-sair da tabela: '))  
  
 print('\n\n')  
 if resps == 0:  
 mostratodasdisc()  
  
 elif resps == 1:  
 while True:  
 try:  
 codigodisc = int(input('Digite o código da disciplina: '))  
 except:  
 codigodisc = int(input('Digite o código da disciplina: '))  
  
  
 if consultardisciplina(codigodisc) == 'cadastrado':  
 print()  
 elif consultardisciplina(codigodisc) == 'naocadastrado':  
 print('Disciplina não cadastrada')  
 break  
 print('\n')  
  
 if resps == 2:  
 while True:  
 try:  
 codigodisc = int(input('Digite o código da disciplina: '))  
 except:  
 codigodisc = int(input('Digite o código da disciplina: '))  
  
  
  
 nomedisciplina = input('Nome da Disciplina: ')  
  
 msg = cadastrardisciplina(codigodisc, nomedisciplina)  
 print(msg)  
 break  
 print('\n')  
 if resps == 3:  
 print('Atenção: Código da disciplina não pode ser alterado: ')  
 while True:  
 try:  
 codigodisc = int(input('Digite o código da disciplina: '))  
 except:  
 codigodisc = int(input('Digite o código da disciplina: '))  
  
  
  
 nomedisciplina = input('Informe novo nome da disciplina: ')  
  
 msg = alterardisciplina(codigodisc, nomedisciplina)  
 print(msg)  
 break  
 print('\n')  
 elif resps == 4:  
 while True:  
 try:  
 codigodisc = int(input('Digite o código da disciplina: '))  
 except:  
 codigodisc = int(input('Digite o código da disciplina: '))  
  
  
  
 confirma = input('Confirme a exclusão S-SIM OU N-NÃO: ')  
  
 while confirma != 'S' and confirma != 'N': confirma = input('Digite S-SIM OU N-NÃO: ')  
 msg = excluirdisciplina(codigodisc)  
 print (msg)  
 break  
  
 elif resps == 5:  
 print('\n')  
 print('='\*80)  
 break  
 print('\n')  
  
  
 elif resposta == 2:  
 if abrebanco() == 1:  
 print('=' \* 80)  
 print('{:^80}'.format('SISTEMA UNIVAP - PROFESSORES'))  
 print('=' \* 80)  
 while True:  
 try:  
 resps = int(input('Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar um professor \n 2-cadastrar um professor \n 3-alterar o nome de um professor \n 4-excluir um professor \n 5-sair da tabela: '))  
 except:  
 resps = int(input('Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar um professor \n 2-cadastrar um professor \n 3-alterar o nome de um professor \n 4-excluir um professor \n 5-sair da tabela: '))  
  
 while (resps < 0) or (resps > 5):  
 try:  
 resps = int(input(  
 'Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar um professor \n 2-cadastrar um professor \n 3-alterar o nome de um professor \n 4-excluir um professor \n 5-sair da tabela: '))  
 except:  
 resps = int(input(  
 'Digite a ação desejada:\n 0-selecionar todos \n 1-selecionar um professor \n 2-cadastrar um professor \n 3-alterar o nome de um professor \n 4-excluir um professor \n 5-sair da tabela: '))  
 print('\n')  
 if resps == 0:  
 mostratodasprof()  
  
 elif resps == 1:  
 while True:  
 try:  
 registro = int(input('Digite o código do professor: '))  
 except:  
 registro = int(input('Digite o código do professor: '))  
  
 if consultarprofessor(registro) == 'cadastrado':  
 print('')  
 elif consultarprofessor(registro) == 'naocadastrado':  
 print('Professor não cadastrado')  
 break  
 if resps == 2:  
 while True:  
 try:  
 registro = int(input('Digite o código do professor: '))  
 except:  
 registro = int(input('Digite o código do professor: '))  
  
 try:  
 nomeprofessor = input('Nome do professor: ')  
 telefoneprof = input('telefone: ')  
 idadeprof = int(input('idade: '))  
 salarioprof = int(input('salário: '))  
 except:  
 nomeprofessor = input('Nome do professor: ')  
 telefoneprof = input('telefone: ')  
 idadeprof = int(input('idade: '))  
 salarioprof = int(input('salário: '))  
  
 while idadeprof > 100:  
 try:  
 idadeprof = int(input('idade: '))  
 except:  
 idadeprof = int(input('idade: '))  
 msg = cadastrarprofessor(registro, nomeprofessor, telefoneprof, idadeprof, salarioprof)  
 print(msg)  
 break  
  
 if resps == 3:  
 print('Atenção: Código do professor não pode ser alterado: ')  
 while True:  
 try:  
 registro = int(input('Digite o código do professor: '))  
 except:  
 registro = int(input('Digite o código do professor: '))  
  
 try:  
 nomeprofessor = input('Nome do professor: ')  
 telefoneprof = input('telefone: ')  
 idadeprof = int(input('idade: '))  
 salarioprof = float(input('salário: '))  
 except:  
 nomeprofessor = input('Nome do professor: ')  
 telefoneprof = input('telefone: ')  
 idadeprof = int(input('idade: '))  
 salarioprof = float(input('salário: '))  
  
 while idadeprof > 100:  
 try:  
 idadeprof = int(input('idade: '))  
 except:  
 idadeprof = int(input('idade: '))  
 msg = alterarprofessor(registro, nomeprofessor, telefoneprof, idadeprof, salarioprof)  
 print(msg)  
 break  
  
 elif resps == 4:  
 while True:  
 try:  
 registro = int(input('Digite o código do professor: '))  
 except:  
 registro = int(input('Digite o código do professor: '))  
  
 confirma = input('Confirme a exclusão S-SIM OU N-NÃO: ')  
  
 while confirma != 'S' and confirma != 'N': confirma = input('Digite S-SIM OU N-NÃO: ')  
 msg = excluirprofessor(registro)  
 print(msg)  
 break  
  
 elif resps == 5:  
 print('=' \* 80)  
 break  
  
  
  
  
 elif resposta == 3:  
 if abrebanco() == 1:  
 print('=' \* 80)  
 print('{:^80}'.format('SISTEMA UNIVAP - DISCIPLINAS X PROFESSORES'))  
 print('=' \* 80)  
 while True:  
 try:  
 resps = int(input('Digite a ação desejada:\n 0-sair da tabela \n 1-selecionar registro \n 2-cadastrar novo registro \n 3-alterar um registro \n 4-excluir um registro \n 5-selecionar todos: '))  
 except:  
 resps = int(input('Digite a ação desejada:\n 0-sair da tabela \n 1-selecionar registro \n 2-cadastrar novo registro \n 3-alterar um registro \n 4-excluir um registro \n 5-selecionar todos: '))  
  
  
 while (resps < 0) or resps > 6:  
 resps = int(input('Digite a ação desejada:\n 0-sair da tabela \n 1-selecionar registro \n 2-cadastrar novo registro \n 3-alterar um registro \n 4-excluir um registro \n 5-selecionar todos: '))  
 print('\n')  
 if resps == 5:  
 mostratodasdiscxprof()  
  
 elif resps == 1:  
 while True:  
 try:  
 cod = int(input('Digite o código do registro: '))  
 except:  
 cod = int(input('Digite o código do registro: '))  
  
 if consultardiscxprof(cod) == 'cadastrado':  
 print()  
 elif consultardiscxprof(cod) == 'naocadastrado':  
 print('registro não está cadastrado')  
 break  
 if resps == 2:  
 while True:  
 confirmar = input('Quer verificar todos os professores e disciplinas antes? S-SIM OU N-NÃO: ')  
 while confirmar != 'S' and confirmar != 'N':  
 confirmar = input('Digite S-SIM OU N-NÃO: ')  
 if confirmar == 'S':  
 mostratodasdisc()  
 mostratodasprof()  
  
 try:  
 cod = int(input('Digite o código do registro: '))  
 except:  
 cod = int(input('Digite o código do registro: '))  
  
 try:  
 coddisc = int(input('código disciplina: '))  
 codprof = int(input('código professor: '))  
 curso = int(input('curso: '))  
 cargahoraria = int(input('carga horária: '))  
 anoletivo = int(input('ano letivo: '))  
 except:  
 coddisc = int(input('código disciplina: '))  
 codprof = int(input('código professor: '))  
 curso = int(input('curso: '))  
 cargahoraria = int(input('carga horária: '))  
 anoletivo = int(input('ano letivo: '))  
  
  
  
  
  
 msg = cadastrardiscxprof(cod, coddisc, codprof, curso, cargahoraria, anoletivo)  
 print(msg)  
 break  
 if resps == 3:  
 print('Atenção: Código do professor não pode ser alterado: ')  
 while True:  
 try:  
 cod = int(input('Digite o código do registro: '))  
 except:  
 cod = int(input('Digite o código do registro: '))  
  
 try:  
 coddisc = int(input('código disciplina: '))  
 codprof = int(input('código professor: '))  
 curso = int(input('curso: '))  
 cargahoraria = int(input('carga horária: '))  
 anoletivo = int(input('ano letivo: '))  
 except:  
 coddisc = int(input('código disciplina: '))  
 codprof = int(input('código professor: '))  
 curso = int(input('curso: '))  
 cargahoraria = int(input('carga horária: '))  
 anoletivo = int(input('ano letivo: '))  
 msg = alterardiscxprof(cod, coddisc, codprof, curso, cargahoraria, anoletivo)  
 print(msg)  
 break  
  
 elif resps == 4:  
 while True:  
 try:  
 cod = int(input('Digite o código do registro: '))  
 except:  
 cod = int(input('Digite o código do registro: '))  
  
 confirma = input('Confirme a exclusão S-SIM OU N-NÃO: ')  
 while confirma != 'S' and confirma != 'N': confirma = input('Digite S-SIM OU N-NÃO: ')  
 msg = excluirdiscxprof(cod)  
 print(msg)  
 break  
  
 elif resps == 0:  
 break  
  
 print('\n\n')  
 print('=' \* 80)  
 else:  
 print('Alguma falha ocorreu na conexão com o bando de dados.')  
  
 elif resposta == 4:  
 if abrebanco() == 1:  
 print('=' \* 80)  
 print('{:^80}'.format('SISTEMA UNIVAP - GERADOR HTML'))  
 print('=' \* 80)  
 while True:  
 try:  
 nomeArquivo = int(input('Digite o código do professor: '))  
 except:  
 nomeArquivo = int(input('Digite o código do professor: '))  
  
 if consultarprofessor(nomeArquivo) == 'cadastrado':  
 print('')  
 gerarHTML(nomeArquivo)  
 elif consultarprofessor(nomeArquivo) == 'naocadastrado':  
 print('Professor não cadastrado')  
 break  
 elif resposta == 5:  
 if abrebanco() == 1:  
 print('=' \* 80)  
 print('{:^80}'.format('SISTEMA UNIVAP - GERADOR PDF'))  
 print('=' \* 80)  
 try:  
 r = int(input('MENU\n-Dados de professor(1) \n-Dados de curso(2):'))  
 except:  
 r = int(input('MENU\n-Dados de professor(1) \n-Dados de curso(2):'))  
 if r == 1:  
 try:  
 profPDF = int(input('Pesquisar por: \n-Nome(1) \n-Registro(2):'))  
 except:  
 profPDF = int(input('Pesquisar por: \n-Nome(1) \n-Registro(2):'))  
 if profPDF == 1:  
 gerarPDF(r,profPDF)  
 elif profPDF == 2:  
 gerarPDF(r,profPDF)  
 elif r == 2:  
 try:  
 cursoPDF = int(input('Pesquisar por: \n-Disciplinas um curso(1) \n-Professores de um curso(2) \n- Carga horária total do curso(3)'))  
 except:  
 cursoPDF = int(input('Pesquisar por: \n-Disciplinas um curso(1) \n-Professores de um curso(2) \n- Carga horária total do curso(3)'))  
 if cursoPDF == 1:  
 gerarPDF(r,cursoPDF)  
 elif cursoPDF == 2:  
 gerarPDF(r,cursoPDF)  
 elif cursoPDF == 3:  
 gerarPDF(r,cursoPDF)  
 elif resposta == 6:  
 print('\n\n')  
 print('=' \* 80)  
 print('obrigado por testar o programa')  
 break  
  
 else:  
 print('Digite um número válido')