[Projeto Final – Trabalho Prático]

[433- G4m3c3nt3r]



Autor(es):

**Hugo Sousa nº81751, Vasco Curado nº81753**

**Rui Caetano nº81741**

Disciplina/Professor:

[Programação Avançada / Fernando Barros]

Academia de Ensino Superior de Mafra, 24 de junho de 2022

Índice

[1 - Introdução ii](#_Toc106974811)

[2 - Desenvolvimento 1](#_Toc106974812)

[2.1. Biblioteca 1](#_Toc106974813)

[2.1.1. Tkinter 1](#_Toc106974814)

[2.1.2. Tkcalendar 1](#_Toc106974815)

[2.1.3. Pyodbc 2](#_Toc106974816)

[2.1.4. Pillow 2](#_Toc106974817)

[2.2. Janela de boas-vindas (Ficheiro main.py) 2](#_Toc106974818)

[2.3. Sistema de registos 3](#_Toc106974819)

[2.4. Sistema de Logins 7](#_Toc106974820)

[2.5. Menu de jogos 9](#_Toc106974821)

[2.6. Explicação do Jogo da Cobra 10](#_Toc106974822)

[2.7. Pontuações 12](#_Toc106974823)

[2.8. Conexão Base Dados (ficheiro Base\_dados.py) 14](#_Toc106974824)

[3 - Pontos a melhorar 15](#_Toc106974825)

# Introdução

Com este projeto tivemos o intuito de criar uma aplicação que consiste num centro de jogos com sistema de logins e de registo. Todos os valores inseridos serão guardados numa base de dados criada previamente por nós (em SQL server). A aplicação foi desenvolvida em Python, com parte gráfica para um uso mais simples ao utilizador.

O programa é constituído por sete diferentes janelas, sendo estas respetivamente:

* Janela de boas-vindas;
* Janela de registo;
* Janela de logins;
* Janela de menu de jogos;
* Janela dedicada à explicação do jogo da cobra;
* Janela do jogo;
* Janela de pontuações.

# Desenvolvimento

Nesta componente de desenvolvimento iremos detalhar o codigo de forma mais acentuada,visto que o mesmo está segmentado em 3 ficheiros .py (main, Base\_dados e jogo\_cobra).

Esta explicação será dividida em 3 partes:

* Bibliotecas;
* Detalhes de cada ecrã;
* Ligação à base de dados.

## Biblioteca

Foram utilizadas cinco diferentes bibliotecas para o funcionamento do programa.

Diferentes bibliotecas:

import tkinter as tk  
from tkcalendar import DateEntry #pip install tkcalendar   
from PIL import ImageTk, Image #pip install pillow

import pyodbc #pip install pyodbc

### Tkinter

Tkinter é uma biblioteca especializada em configuração de parte gráfica. Foi com esta que criamos as diferentes janelas que constituem o nosso projeto, sendo a que está maioritariamente presente no nosso código.

### Tkcalendar

Importado apenas a componente DateEntry, para um dos campos de registo (data de nascimento).

### Pyodbc

Foi utlizada para fazer a conexão à base de dados e todas as outras componentes relacionadas à mesma, tais como inserir valores de registo e verificar a existência de informação na base de dados.

### Pillow

Importadas apenas as componentes de ImageTk e Image para o uso de imagens previamente inseridas numa pasta localizada no ficheiro do projeto.

## Janela de boas-vindas (Ficheiro main.py)

Primeiro e principal ecrã da nossa aplicação. Tem como objetivo dar umas boas-vindas ao utilizador e permitir a opção de iniciar sessão ou de criar uma nova conta.

Código:

window = tk.Tk()  
window.geometry("800x600")  
window.resizable(0,0) #impede a alteração do tamanho da janela  
imagem1 = Image.open('imagens\\bg\_janela1.jpeg')  
bg1 = ImageTk.PhotoImage(imagem1)  
bg = tk.Label(window, image=bg1)  
bg.image = imagem1  
bg.pack()

window.title("433 G4M3C3NT3R")  
tk.Button(window, text="Iniciar sessão", font=("",15), command=login).place(x=210,y=370)  
tk.Button(window, text="Criar conta", font=("",15), command=register2).place(x=475,y=370)  
window.mainloop()

O mesmo irá criar uma janela com uma imagem de fundo e dois botões, em que um é responsável por levar para a página de iniciar sessão e outro para a página de criar conta.

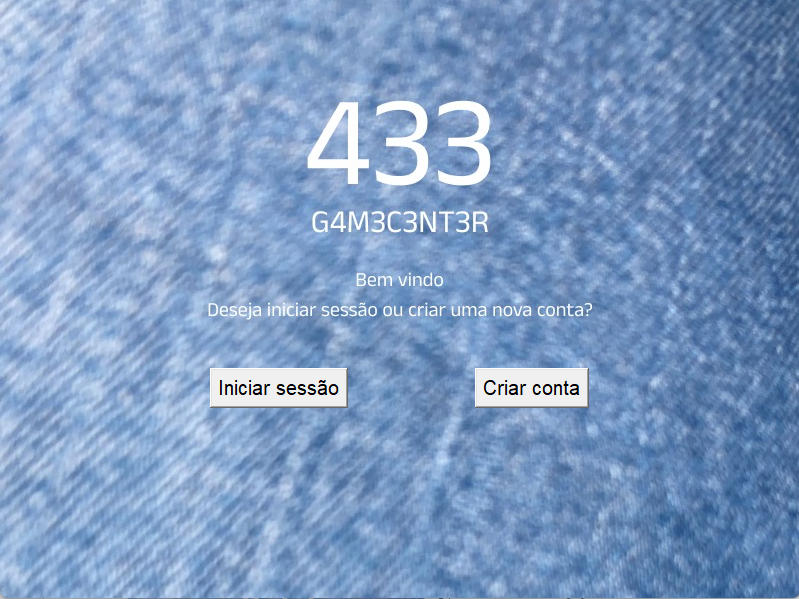
Funcionalidade do botão iniciar sessão:

def login():  
 window.destroy()  
 ecra2()

Funcionalidade do botão criar conta:

def register2():  
 window.destroy()  
 register()

Execução do programa:



## Sistema de registos

Esta janela destinasse à criação de uma conta para novos utilizadores, sendo ela composta por uma imagem de fundo e widgets de caixas de texto para inserir a informação dos mesmos.

Código:

window2 = tk.Tk()  
window2.geometry("1280x720")  
window2.resizable(0,0) #impede a alteração do tamanho da janela  
imagem2 = Image.open('imagens\\bg\_janela2.jpeg')  
imagem2 = ImageTk.PhotoImage(imagem2)  
bg2 = tk.Label(window2, image=imagem2)  
bg2.image = imagem1  
bg2.pack()

window2.title("433 G4M3C3NT3R - Criar conta")  
tk.Label(window2, text="Preencha os seguintes campos para criar uma conta",font=("", 15), bg="white").place(x=750, y=50)  
tk.Label(window2, text="Nome:", font=("", 15), bg="white").place(x=730, y=110)  
nome = tk.Entry(window2, width=25)  
nome.place(x=792, y=117)  
nomeerro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
nomeerro.place(x=730, y=140)  
tk.Label(window2, text="Apelido:", font=("", 15), bg="white").place(x=970, y=110)  
apelido = tk.Entry(window2, width=25)  
apelido.place(x=1045, y=117)  
apelidoerro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
apelidoerro.place(x=970, y=140)  
tk.Label(window2, text="Nome de utilizador:", font=("", 15), bg="white").place(x=730, y=160)  
nickname = tk.Entry(window2, width=25)  
nickname.place(x=902, y=167)  
nickerro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
nickerro.place(x=730, y=190)  
tk.Label(window2, text="Email:", font=("", 15), bg="white").place(x=730, y=220)  
email = tk.Entry(window2, width=35)  
email.place(x=787, y=225)  
emailerro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
emailerro.place(x=730, y=250)  
tk.Label(window2, text="Data de nascimento:", font=("", 15), bg="white").place(x=730, y=270)  
data = DateEntry(window2,width=25)  
data.place(x=917, y=275)  
tk.Label(window2, text="Palavra-passe:", font=("", 15), bg="white").place(x=730, y=300)  
pass\_str = tk.StringVar()  
confirm = tk.StringVar()  
password = tk.Entry(window2, width=25, show="\*", textvariable=pass\_str)  
password.place(x=866, y=306)  
passerro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
passerro.place(x=740, y=330)  
tk.Label(window2, text="Confirme:", font=("", 15), bg="white").place(x=1030, y=300)  
confirmpass = tk.Entry(window2, width=25, show="\*", textvariable=confirm)  
confirmpass.place(x=1119, y=306)  
passconferro = tk.Label(window2, text="", font=("", 8), fg="red", bg="white")  
passconferro.place(x=1030, y=330)

a = tk.IntVar(value=0)

mostarpass = tk.Checkbutton(window2, text="Mostrar palavra-passe", variable=a, onvalue=1, offvalue=0, command=mostarpass)  
mostarpass.place(x=730, y=350)  
confirmpasserro = tk.Label(window2, text="", font=("", 9), fg="red", bg="white")  
confirmpasserro.place(x=730, y=330)  
criadosucesso = tk.Label(window2, text="", font=("", 9), fg="green", bg="white")  
criadosucesso.place(x=820, y=430)

tk.Button(window2, text="Criar conta", font=("", 15),command=Inserir\_BD).place(x=940, y=460)  
tk.Button(window2, text="iniciar sessão em alternativa", font=("", 15), command=login2).place(x=860, y=500)  
window2.mainloop()

Funcionalidade do botão iniciar sessão em alternativa:

def login2():  
 window2.destroy()  
 ecra2()

Funcionalidade do checkbutton (mostrar palavra-passe):

def mostarpass():  
 if (a.get()==1):  
 password.config(show='')  
 confirmpass.config(show='')  
 else:  
 password.config(show='\*')  
 confirmpass.config(show='\*')

Botão criar conta:

Este botão terá uma função mais extensa visto que ao ser pressionado irá verificar se não existe nenhum erro no preenchimento dos campos como também enviará os valores para a base dados

def Inserir\_BD():  
 a, b, c, d, e, f = len(nome.get()), len(apelido.get()), len(nickname.get()), len(email.get()), len(password.get()), len(confirmpass.get())  
 # Erro1 - nome não foi inserido  
 if a == 0:  
 nomeerro.config(text="O nome não foi inserido")  
 else:  
 nomeerro.config(text="")  
 # Erro2 - apelido não foi inserido  
 if b == 0:  
 apelidoerro.config(text="O apelido não foi inserido")  
 else:  
 apelidoerro.config(text="")  
 # Erro3 - username não foi inserido  
 if c == 0:  
 nickerro.config(text="O nome de utilizador não foi inserido")  
 else:  
 nickerro.config(text="")  
 # Erro4 - email não foi inserido  
 if d == 0:  
 emailerro.config(text="O email não foi inserido")  
 else:  
 emailerro.config(text="")  
 # Erro5 - password não foi inserido  
 if e == 0:  
 passerro.config(text="A palavra-passe não foi inserida")  
 else:  
 passerro.config(text="")  
 # Erro6 - confirmar password não foi inserido  
 if f == 0:  
 passconferro.config(text="A confirmação de palavra-passe não foi inserida")  
 else:  
 passconferro.config(text="")  
 # Erro7 - password e confirmar password são diferentes  
 if (e!=0) and (f!=0):  
 if password.get() != confirmpass.get():  
 confirmpasserro.config(text="A palavra-passe não é igual")  
 else:  
 confirmpasserro.config(text="")  
  
 if (a!=0) and (b!=0) and (c!=0) and (d!=0) and (e!=0) and (f!=0) and (password.get() == confirmpass.get()):  
 try:  
 print(nome.get())  
 cursor = conexao\_sql()  
 comando = f'''INSERT INTO users(nome, apelido, nickname, email,password,data)  
 VALUES ('{nome.get()}','{apelido.get()}','{nickname.get()}','{email.get()}','{password.get()}','{data.get()}')'''  
 cursor.execute(comando)  
 cursor.commit()  
 criadosucesso.config(text='Conta criada com Sucesso. Clique em login para iniciar sessão')  
 except: # pois o campo nickname na base dados é unico  
 nickerro.config(text='Nome de utilizador já utilizado')

Execução do programa:

Uma imagem com texto

Descrição gerada automaticamente

## Sistema de Logins

Este ecrã tem como objetivo dar a possibilidade ao utilizador de fazer o seu login, sendo o mesmo composto por os campos dos logins propriamente ditos e dois botões, um criar conta em que o utilizador será direcionado para a pagina de criar conta e outro para entrar na aplicação.

window3 =tk.Tk()  
window3.geometry("1280x720")  
window3.resizable(0,0)  
imagem3 = Image.open('imagens\\bg\_janela3.jpeg')  
imagem3 = ImageTk.PhotoImage(imagem3)  
bg3 = tk.Label(window3, image=imagem3)  
bg3.image = imagem3  
bg3.pack()  
window3.title("433 G4M3C3NT3R - Iniciar sessão")

tk.Label(window3, text="Utilizador:", font=(" ", 25), bg="white").place(x=800, y=120)  
nickname = tk.Entry(window3, width=25)  
nickname.place(x=950, y=135)  
erronick = tk.Label(window3, text="", font=("", 9), fg="red", bg="white")  
erronick.place(x=800, y=170)  
tk.Label(window3, text="Palavra-passe:", font=("", 25), bg="white").place(x=750, y=250)  
password = tk.Entry(window3, width=25, show="\*")  
password.place(x=970, y=265)  
erropass = tk.Label(window3, text="", font=("", 9), fg="red", bg="white")  
erropass.place(x=750, y=300)  
semsucesso = tk.Label(window3, text='', font=('', 9), fg='red', bg="white")  
semsucesso.place(x=860, y=350)  
a = tk.IntVar(value=0)

mostarpass = tk.Checkbutton(window3, text="Mostrar palavra-passe", variable=a, onvalue=1, offvalue=0, command=mostarpass)  
mostarpass.place(x=970, y=300)

tk.Button(window3, text="Entrar", font=("", 20), command=Consultar\_BD).place(x=820, y=400)  
tk.Button(window3, text="Criar conta", font=("", 20), command=criarconta2).place(x=960, y=400)  
window3.mainloop()

Funcionalidade do checkbutton(mostrar palavra-passe):

def mostarpass():  
 if (a.get()==1):  
 password.config(show='')  
 else:  
 password.config(show='\*')

Funcionalidade do botão criar conta:

def criarconta2():  
 window3.destroy()  
 register()

Funcionalidade do botao Entrar:

Semelhante a um botão anterior este também irá ter uma funcionalidade mais extensa, pois o mesmo terá de verificar se não ocorreu nenhum erro de preenchimento e se tanto o utilizador como a palavra-passe correspondem na Base de Dados.

def Consultar\_BD():  
 a, b = len(nickname.get()), len(password.get())  
 # Erro1 - username não foi inserido  
 if a == 0:  
 erronick.config(text="O nome de utilizador não foi inserido")  
 else:  
 erronick.config(text="")  
 # Erro2 - password não foi inserido  
 if b == 0:  
 erropass.config(text="A palavra-passe não foi inserida")  
 else:  
 erropass.config(text="")  
 #código que verifica se existe o username e a password  
 if (a != 0) and (b != 0):  
 var1 = (nickname.get())  
 var2 = (password.get())  
 cursor = conexao\_sql()  
 comando = """EXEC logins @nickname=?, @password=?""" #stored procedure criado na base dados  
 cursor.execute(comando, var1, var2)  
 resultado = cursor.fetchall()  
 if resultado:  
 window3.destroy()  
 menujogos()  
 else:  
 semsucesso.config(text='User ou Password nao correspondem')

Execução do programa:



## Menu de jogos

Esta janela é responsável por mostrar os diferentes jogos disponíveis no nosso programa. O ecrã tem uma imagem de fundo e três botões: um para aceder a uma página do jogo da cobra, outro sem funcionalidade para exemplificar a posição de outra pagina de jogo e um botão para encerrar o programa.

window4 = tk.Tk()  
window4.geometry("1280x720")  
window4.resizable(0, 0)  
imagem4 = Image.open('imagens\\bg\_janela4.jpeg')  
bg4 = ImageTk.PhotoImage(imagem4)  
bg = tk.Label(window4, image=bg4)  
bg.image = imagem4  
bg.pack()  
window4.title("433 G4M3C3NT3R - Menu de jogos")

tk.Label(window4, text="Menu de Jogos", font=("", 20), bg="white").place(x=540, y=30)  
tk.Button(window4, text=("Jogo da Cobra"), font=("", 12),command=jogocobra, width=55, height=15).place(x=90, y=90)  
frame2 = tk.LabelFrame(window4)  
frame2.place(x=650, y=90)  
tk.Button(frame2, text=("Em breve..."), font=("", 12), width=55, height=15).pack()  
tk.Button(window4, text=("Sair"), font=("", 20), width=10, bg='#EE0000', command=window4.destroy).place(x=1000,y=500)  
window4.mainloop()

Funcionalidade do Botão(jogo da cobra):

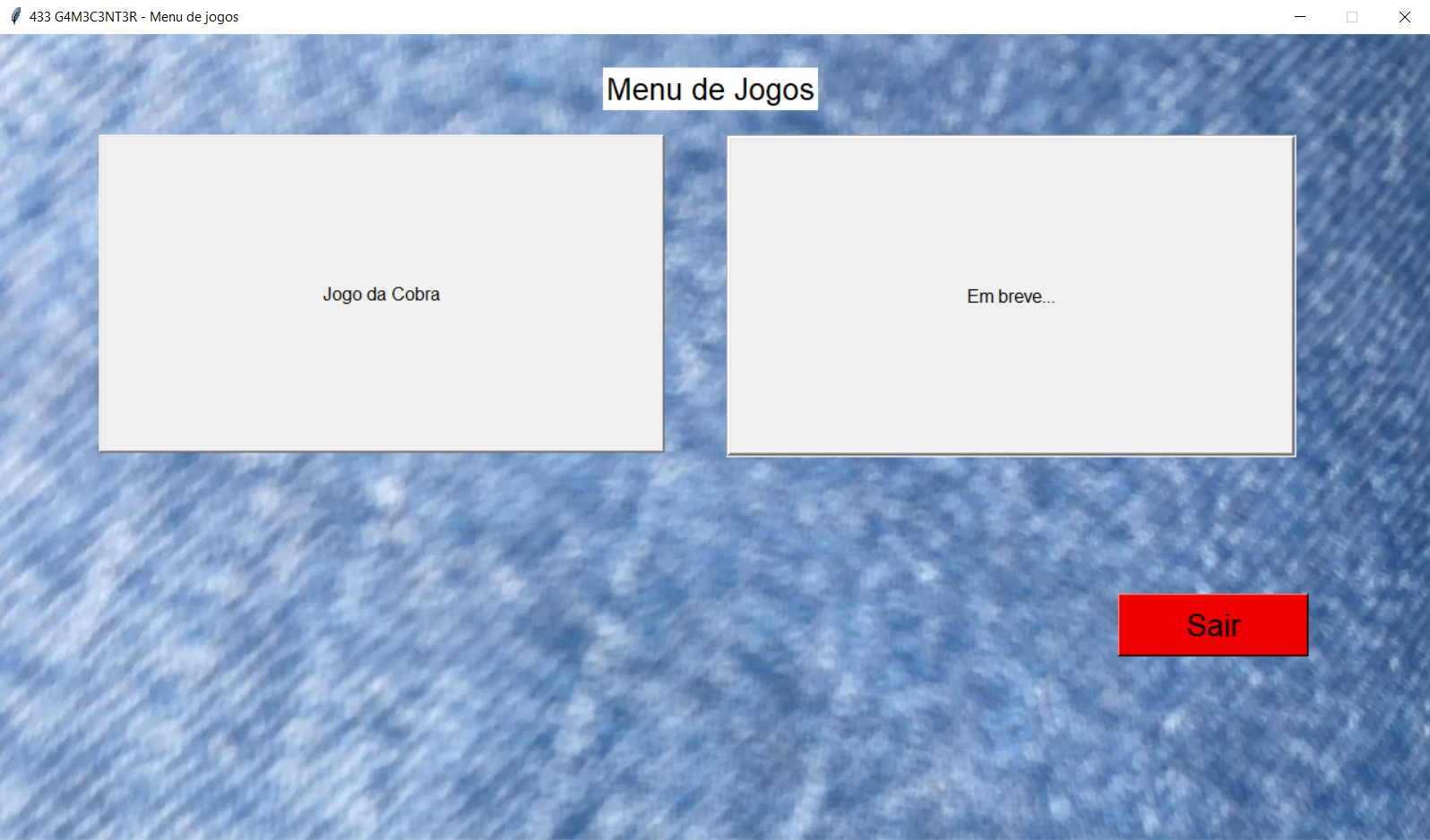
def jogocobra():  
 window4.destroy()  
 janela\_jogocobra()

Funcionalidade do Botão Sair:

Como apenas sai da aplicação não é preciso adicionar nenhuma função extra ao comando

command=window4.destroy

Execução do programa:



## Explicação do Jogo da Cobra

Esta janela é dedicada para a explicação do jogo, a mesma é composta pela tal explicação e três botões. Um botão voltar em que irá para o menu de jogos outra vez, um botão iniciar que dará inicio ao jogo e um botão pontuações que abrirá outra janela.

window5 = tk.Tk()  
window5.geometry("1280x720")  
window5.resizable(0, 0)  
imagem5 = Image.open('imagens\\bg\_janela5.jpeg')  
bg5 = ImageTk.PhotoImage(imagem5)  
bg = tk.Label(window5, image=bg5)  
bg.image = imagem5  
bg.pack()  
window5.title("433 G4M3C3NT3R - Jogo da Cobra")

tk.Button(window5, text="Voltar", font=("", 15),command=menujogos2, bg="white").place(x=50, y=30)  
tk.Label(window5, text="Jogo da cobra", font=("", 20), bg="white").place(x=900, y=30)  
tk.Label(window5, text="ID Jogo: 2", font=("", 15), bg="white").place(x=720, y=120)  
tk.Label(window5, text="Como jogar?", font=("", 15), bg="white").place(x=720, y=150)  
tk.Label(window5, text="Movimentação:", font=("", 15), bg="white").place(x=720, y=190)  
tk.Label(window5, text="Up Arrow - mover para cima", font=("", 15), bg="white").place(x=720, y=220)  
tk.Label(window5, text="Down Arrow - mover para baixo", font=("", 15), bg="white").place(x=720, y=250)  
tk.Label(window5, text="Left Arrow - mover para a esquerda", font=("", 15), bg="white").place(x=720, y=280)  
tk.Label(window5, text="Right Arrow - mover para a direita", font=("", 15), bg="white").place(x=720, y=310)  
tk.Label(window5, text="Objetivo:", font=("",15), bg="white").place(x=720, y=370)  
tk.Label(window5, text="Com a cobra comer o máximo de número de maçãs,\n sem tocar nas bordas do mapa e nem em sim própria.\n Quantas mais maçãs forem comidas, maior é a pontuação.\n Bom Jogo!\n Premir Pontuações para adicionar o seu melhor resultado!", font=("", 15), bg="white").place(x=700, y=400)

tk.Button(window5, text="Iniciar", font=("", 15), command=jogo). place(x=920, y=600)  
tk.Button(window5, text="Pontuações", font=("", 15), command=pontos).place(x=900, y=550)  
window5.mainloop()

Funcionalidade Botão Voltar:

def menujogos2():  
 window5.destroy()  
 menujogos()

Funcionalidade Botão Iniciar:

def jogo():  
 from jogo\_cobra import jogar  
 jogar()

Execução do programa:

Uma imagem com texto

Descrição gerada automaticamente

## Pontuações

Janela com o objetivo de o utilizador adicionar a pontuação que fez no jogo. É composta por uma tabela que mostra o utilizador com mais pontos no jogo, três campos para preencher e um botão para enviar os valores para base de dados depois de inseridos.

window6 = tk.Tk()  
window6.geometry("500x620")  
window6.resizable(0, 0)  
window6.title("433 G4M3C3NT3R - Pontuações")  
tk.Label(window6, text="HighScore", font=("", 20)).place(x=30, y=30)  
id\_jogo = tk.Entry(window6, width=25)  
id\_jogo.place(x=200, y=285)  
erroidjogo = tk.Label(window6, text="", font=("", 9), fg="red")  
erroidjogo.place(x=200, y=320)  
nickname = tk.Entry(window6, width=25)  
nickname.place(x=200, y=365)  
erronick = tk.Label(window6, text="", font=("", 9), fg="red")  
erronick.place(x=200, y=400)  
pontuacao = tk.Entry(window6, width=25)  
pontuacao.place(x=200, y=445)  
erropontuacao = tk.Label(window6, text="", font=("", 9), fg="red")  
erropontuacao.place(x=200, y=480)  
inseridosucesso = tk.Label(window6, text="", font=("", 9), fg="green")  
inseridosucesso.place(x=125, y=560)

a=top1()  
tk.Label(window6, text=a, font=("", 15)).place(x=100, y=100)  
tk.Button(window6, text="Inserir Pontos", font=("", 15),command=Inserir\_BD2).place(x=150, y=520)  
tk.Label(window6, text="Jogo:", font=("", 15)).place(x=100, y=280)  
tk.Label(window6, text="Nickname:", font=("", 15)).place(x=100, y=360)  
tk.Label(window6, text="Pontos:", font=("", 15)).place(x=100, y=440)

Funcionalidade do Botão Inserir Pontos:

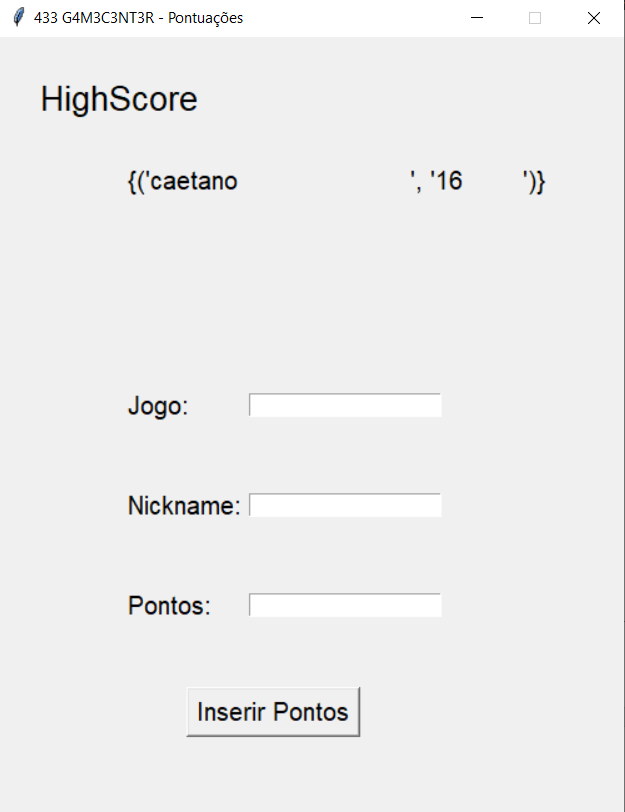
Visto que se tem de verificar se os campos estão preenchidos e se o utilizador existe na base dados antes de inserir teve-se de criar uma função para tal

def Inserir\_BD2():  
 a, b, c = len(nickname.get()), len(pontuacao.get()), len(id\_jogo.get())  
 # Erro1 - nickname não foi inserido  
 if a == 0:  
 erronick.config(text="O nickname não foi inserido")  
 else:  
 erronick.config(text="")  
 # Erro2 - password não foi inserido  
 if b == 0:  
 erropontuacao.config(text="A Pontuação não foi inserida")  
 else:  
 erropontuacao.config(text="")  
 if c == 0:  
 erroidjogo.config(text="O id do jogo nao foi inserido")  
 else:  
 erroidjogo.config(text="")  
  
 if (a != 0) and (b != 0) and (c != 0):  
 try:  
 cursor = conexao\_sql()  
 comando = f'''INSERT INTO pontuacao(nickname,id\_jogo,pontuacao)  
 VALUES ('{nickname.get()}','{id\_jogo.get()}','{pontuacao.get()}')'''  
 cursor.execute(comando)  
 cursor.commit()  
 inseridosucesso.config(text='Pontuação inserida com sucesso')  
 except:  
 erronick.config(text="O nickname não existe na Base de Bados")

Tabela de Highscore:

def top1():  
 cursor = conexao\_sql()  
 comando = '''SELECT TOP 1 nickname,pontuacao from pontuacao ORDER BY pontuacao DESC'''  
 cursor.execute(comando)  
 resultado=cursor.fetchall()  
 cursor.commit()  
 return resultado  
a=top1()  
tk.Label(window6, text=a, font=("", 15)).place(x=100, y=100)

Execução do programa:



## Conexão Base Dados (ficheiro Base\_dados.py)

Este ficheiro é utilizado único e exclusivamente para fazer a ligação ao servidor SQL onde foi criado uma base dados chamada ‘PA\_81753\_81741\_81751’ com os seguintes tabelas:

* Users - id\_user, nome, apelido, nickname, email, data e password.
* Jogos – id\_jogo, nome, descrição, tipo, idade\_restrição e data\_lançamento.
* Pontuacao – nickname, id\_jogo, pontuação.

Para fazer a tal conexão usamos o seguinte código:

import pyodbc #pip install pyodbc  
  
  
#Funçao de conexao  
def conexao\_sql():  
 server = 'srvsql-ipt.ddns.net'  
 database ='PA\_81753\_81741\_81751'  
 username = '81753'  
 password = '81753'  
 string\_conexao='Driver={SQL Server Native Client 11.0};Server='+server+';Database='+database+';UID='+username+';PWD='+password  
 conexao = pyodbc.connect(string\_conexao)  
  
 return conexao.cursor()

'srvsql-ipt.ddns.net'- é o nome do servidor em que a base dados está inserida

'PA\_81753\_81741\_81751' – nome da base dados em si

'81753' – credenciais de utilizador e password para entrar no servidor já criado anteriormente

# Pontos a melhorar

Reconhecemos que alguns dos aspetos a melhorar no nosso projeto passam por:

* Login com face id (reconhecimento facial);
* Inserção automática dos pontos obtidos no jogo, ou seja, quando perdemos o programa guardar os pontos para posteriormente os inserir na base de dados;
* Melhorar o design da Leaderboard (tabela de pontuações);
* Desenvolver o nosso próprio jogo;
* Apresentar mais jogos;

# Conclusão

Com o desenvolvimento deste trabalho concluímos que existe muito trabalho por de trás dos projetos desenvolvidos em python. Desde escolher a dedo as bibliotecas que necessitamos até fazer os ajustes necessários para ficar tudo alinhado.

Em geral conseguimos ultrapassar as nossas dificuldades, com o auxílio uns dos outros, porem não estamos 100% satisfeitos com o resultado. Temos noção que com mais tempo, teríamos desenvolvido melhor e com mais calma o projeto.