Tela de login

import tkinter as tk

from tkinter import messagebox

def verificar\_login():

usuario = entry\_usuario.get()

senha = entry\_senha.get()

# Exemplo de verificação

if usuario == "admin" and senha == "senha123":

messagebox.showinfo("Login", "Login bem-sucedido!")

# Aqui você pode redirecionar para outra tela ou função

else:

messagebox.showerror("Login", "Usuário ou senha incorretos.")

# Criar a janela principal

janela = tk.Tk()

janela.title("Tela de Login")

# Criar widgets

label\_usuario = tk.Label(janela, text="Usuário:")

label\_usuario.pack(pady=5)

entry\_usuario = tk.Entry(janela)

entry\_usuario.pack(pady=5)

label\_senha = tk.Label(janela, text="Senha:")

label\_senha.pack(pady=5)

entry\_senha = tk.Entry(janela, show="\*")

entry\_senha.pack(pady=5)

botao\_login = tk.Button(janela, text="Login", command=verificar\_login)

botao\_login.pack(pady=20)

# Iniciar o loop da interface

janela.mainloop()

Banco Mysql

-- Criar banco de dados

CREATE DATABASE dbEscola;

-- Selecionar o banco de dados

USE dbEscola;

-- Tabela de Cidades

CREATE TABLE cidades (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

estado VARCHAR(100) NOT NULL

);

-- Tabela de Cursos

CREATE TABLE cursos (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

duracao INT NOT NULL, -- Duração em meses

id\_cidade INT,

FOREIGN KEY (id\_cidade) REFERENCES cidades(id)

);

-- Tabela de Alunos

CREATE TABLE alunos (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

data\_nascimento DATE NOT NULL,

id\_cidade INT,

FOREIGN KEY (id\_cidade) REFERENCES cidades(id)

);

-- Tabela de Professores

CREATE TABLE professores (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

especialidade VARCHAR(100) NOT NULL,

id\_cidade INT,

FOREIGN KEY (id\_cidade) REFERENCES cidades(id)

);

-- Tabela de Usuários

CREATE TABLE usuarios (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

senha VARCHAR(255) NOT NULL,

tipo ENUM('admin', 'professor', 'aluno') NOT NULL

);

-- Inserindo cidades

INSERT INTO cidades (nome, estado) VALUES

('Curitiba', 'PR'),

('Porto Alegre', 'RS'),

('Salvador', 'BA');

-- Inserindo cursos

INSERT INTO cursos (nome, duracao, id\_cidade) VALUES

('Engenharia de Software', 48, 1), -- Curitiba

('Administração', 36, 2), -- Porto Alegre

('Design Gráfico', 24, 3); -- Salvador

-- Inserindo alunos

INSERT INTO alunos (nome, data\_nascimento, id\_cidade) VALUES

('Lucas Ferreira', '2005-01-12', 1), -- Curitiba

('Juliana Costa', '2004-08-22', 2), -- Porto Alegre

('Renan Almeida', '2006-05-30', 3); -- Salvador

-- Inserindo professores

INSERT INTO professores (nome, especialidade, id\_cidade) VALUES

('Fernanda Martins', 'Engenharia', 1), -- Curitiba

('Carlos Santos', 'Administração', 2), -- Porto Alegre

('Tatiane Lima', 'Design', 3); -- Salvador

-- Inserindo usuários

INSERT INTO usuarios (nome, email, senha, tipo) VALUES

('superadmin', 'admin@escola.com', 'senhaSegura123', 'admin'),

('professorLucas', 'lucas@escola.com', 'senhaProf456', 'professor'),

('alunaJuliana', 'juliana@escola.com', 'senhaAluno789', 'aluno');

Aluno

import tkinter as tk

from tkinter import messagebox

import mysql.connector

# Função para inserir dados no banco de dados

def inserir\_aluno():

nome = nome\_entry.get()

data\_nascimento = data\_nascimento\_entry.get()

id\_cidade = id\_cidade\_entry.get()

if nome and data\_nascimento:

try:

# Conectar ao banco de dados MySQL

conexao = mysql.connector.connect(

host="localhost", # Substitua pelo host do seu banco

user="seu\_usuario", # Substitua pelo seu usuário do MySQL

password="sua\_senha", # Substitua pela sua senha

database="sua\_base\_de\_dados" # Substitua pelo nome do banco de dados

)

cursor = conexao.cursor()

# Query de inserção

query = "INSERT INTO alunos (nome, data\_nascimento, id\_cidade) VALUES (%s, %s, %s)"

valores = (nome, data\_nascimento, id\_cidade)

# Executar a query

cursor.execute(query, valores)

conexao.commit()

messagebox.showinfo("Sucesso", "Aluno inserido com sucesso!")

except mysql.connector.Error as err:

messagebox.showerror("Erro", f"Erro ao inserir aluno: {err}")

finally:

# Fechar a conexão

if conexao.is\_connected():

cursor.close()

conexao.close()

else:

messagebox.showwarning("Atenção", "Preencha todos os campos!")

# Criar a interface gráfica

janela = tk.Tk()

janela.title("Cadastro de Alunos")

# Label e Entry para o nome

tk.Label(janela, text="Nome").grid(row=0, column=0)

nome\_entry = tk.Entry(janela)

nome\_entry.grid(row=0, column=1)

# Label e Entry para a data de nascimento

tk.Label(janela, text="Data de Nascimento (AAAA-MM-DD)").grid(row=1, column=0)

data\_nascimento\_entry = tk.Entry(janela)

data\_nascimento\_entry.grid(row=1, column=1)

# Label e Entry para o ID da cidade

tk.Label(janela, text="ID da Cidade").grid(row=2, column=0)

id\_cidade\_entry = tk.Entry(janela)

id\_cidade\_entry.grid(row=2, column=1)

# Botão para inserir o aluno

inserir\_btn = tk.Button(janela, text="Inserir Aluno", command=inserir\_aluno)

inserir\_btn.grid(row=3, column=1)

# Iniciar o loop da interface gráfica

janela.mainloop()

cidades

import sqlite3

from tkinter import \*

from tkinter import messagebox

# Conectar ao banco de dados SQLite (ou criar, se não existir)

conn = sqlite3.connect('cidades.db')

cursor = conn.cursor()

# Criar a tabela 'cidades', caso ainda não exista

cursor.execute('''

CREATE TABLE IF NOT EXISTS cidades (

id INTEGER PRIMARY KEY AUTOINCREMENT,

nome TEXT NOT NULL,

estado TEXT NOT NULL

)

''')

conn.commit()

# Função para adicionar uma cidade no banco de dados

def adicionar\_cidade():

nome = entry\_nome.get()

estado = entry\_estado.get()

if nome and estado:

cursor.execute("INSERT INTO cidades (nome, estado) VALUES (?, ?)", (nome, estado))

conn.commit()

messagebox.showinfo("Sucesso", "Cidade adicionada com sucesso!")

entry\_nome.delete(0, END)

entry\_estado.delete(0, END)

carregar\_cidades()

else:

messagebox.showwarning("Erro", "Preencha todos os campos.")

# Função para carregar as cidades do banco de dados

def carregar\_cidades():

cursor.execute("SELECT \* FROM cidades")

registros = cursor.fetchall()

# Limpar a lista antes de atualizar

listbox\_cidades.delete(0, END)

for registro in registros:

listbox\_cidades.insert(END, f"ID: {registro[0]} | Nome: {registro[1]} | Estado: {registro[2]}")

# Criar janela principal

janela = Tk()

janela.title("Cadastro de Cidades")

# Elementos da interface gráfica

Label(janela, text="Nome da Cidade:").grid(row=0, column=0)

entry\_nome = Entry(janela)

entry\_nome.grid(row=0, column=1)

Label(janela, text="Estado:").grid(row=1, column=0)

entry\_estado = Entry(janela)

entry\_estado.grid(row=1, column=1)

botao\_adicionar = Button(janela, text="Adicionar Cidade", command=adicionar\_cidade)

botao\_adicionar.grid(row=2, column=0, columnspan=2)

# Lista para exibir as cidades

listbox\_cidades = Listbox(janela, width=50)

listbox\_cidades.grid(row=3, column=0, columnspan=2)

# Carregar as cidades na lista ao iniciar

carregar\_cidades()

# Iniciar o loop da interface gráfica

janela.mainloop()

# Fechar a conexão com o banco de dados ao sair

conn.close()

usuários

import tkinter as tk

from tkinter import messagebox

import mysql.connector

# Configuração do banco de dados MySQL

db\_config = {

'host': 'localhost',

'user': 'seu\_usuario',

'password': 'sua\_senha',

'database': 'seu\_banco\_de\_dados'

}

# Função para conectar ao banco de dados

def get\_db\_connection():

conn = mysql.connector.connect(\*\*db\_config)

return conn

# Função para listar usuários

def listar\_usuarios():

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute("SELECT \* FROM usuarios")

rows = cursor.fetchall()

conn.close()

lista.delete(0, tk.END) # Limpa a lista de usuários

for row in rows:

lista.insert(tk.END, f"ID: {row[0]}, Nome: {row[1]}, Email: {row[2]}, Tipo: {row[4]}")

# Função para adicionar usuário

def adicionar\_usuario():

nome = entry\_nome.get()

email = entry\_email.get()

senha = entry\_senha.get()

tipo = entry\_tipo.get()

if nome and email and senha and tipo:

conn = get\_db\_connection()

cursor = conn.cursor()

try:

cursor.execute("INSERT INTO usuarios (nome, email, senha, tipo) VALUES (%s, %s, %s, %s)",

(nome, email, senha, tipo))

conn.commit()

messagebox.showinfo("Sucesso", "Usuário adicionado com sucesso!")

listar\_usuarios() # Atualiza a lista de usuários

except mysql.connector.Error as err:

messagebox.showerror("Erro", f"Erro ao adicionar usuário: {err}")

finally:

conn.close()

else:

messagebox.showwarning("Entrada inválida", "Todos os campos são obrigatórios.")

# Função para criar a janela

def criar\_janela():

global entry\_nome, entry\_email, entry\_senha, entry\_tipo, lista

# Janela principal

root = tk.Tk()

root.title("Gerenciador de Usuários")

# Labels e entradas

tk.Label(root, text="Nome").grid(row=0, column=0)

entry\_nome = tk.Entry(root)

entry\_nome.grid(row=0, column=1)

tk.Label(root, text="Email").grid(row=1, column=0)

entry\_email = tk.Entry(root)

entry\_email.grid(row=1, column=1)

tk.Label(root, text="Senha").grid(row=2, column=0)

entry\_senha = tk.Entry(root, show="\*")

entry\_senha.grid(row=2, column=1)

tk.Label(root, text="Tipo").grid(row=3, column=0)

entry\_tipo = tk.Entry(root)

entry\_tipo.grid(row=3, column=1)

# Botão para adicionar usuário

btn\_add = tk.Button(root, text="Adicionar Usuário", command=adicionar\_usuario)

btn\_add.grid(row=4, column=0, columnspan=2)

# Lista de usuários

lista = tk.Listbox(root, width=50)

lista.grid(row=5, column=0, columnspan=2)

# Botão para listar usuários

btn\_list = tk.Button(root, text="Listar Usuários", command=listar\_usuarios)

btn\_list.grid(row=6, column=0, columnspan=2)

# Inicia a janela

root.mainloop()

# Executa a aplicação

criar\_janela()