Examen Practico



Taller de Base de Datos

ING. RUBÉN ALONSO HERNÁNDEZ CHÁVEZ

Isai Guilbert Gordillo

Daniela Alejandra Hernández Rojas

Cesar Emmanuel Rubio Talavera

El código se divide en varias funciones y procedimientos para realizar diferentes tareas:

center\_window(window): Esta función toma una ventana como argumento y la centra en la pantalla del usuario.

agregar\_objeto\_perdido(nombre, categoria, descripcion): Esta función se encarga de agregar un nuevo objeto perdido a la base de datos. Conecta a la base de datos MySQL utilizando los parámetros de conexión proporcionados y ejecuta una consulta INSERT para agregar los valores del nombre, categoría y descripción del objeto perdido. Luego, cierra la conexión a la base de datos.

eliminar\_objeto\_perdido(id\_objeto): Esta función elimina un objeto perdido de la base de datos según su ID. Al igual que en la función anterior, se conecta a la base de datos y ejecuta una consulta DELETE para eliminar el objeto con el ID proporcionado. Luego, cierra la conexión a la base de datos.

consultar\_objetos\_perdidos(): Esta función realiza una consulta SELECT a la base de datos para obtener todos los objetos perdidos almacenados. Luego, crea una nueva ventana utilizando Tkinter y muestra los resultados en una tabla utilizando el widget Treeview. Además, incluye un botón "Eliminar" que permite eliminar objetos seleccionados.

abrir\_ventana\_objetos\_perdidos(): Esta función crea una nueva ventana para administrar los objetos perdidos. La ventana contiene campos para agregar objetos perdidos, así como botones para agregar y regresar al menú principal. Al hacer clic en el botón "Agregar", se llama a la función agregar\_objeto\_perdido() para agregar un nuevo objeto a la base de datos.

regresar\_al\_menu(): Esta función se utiliza para cerrar la ventana de "Objetos perdidos" y regresar al menú principal.

main\_menu(): Esta función es el punto de entrada principal de la aplicación. Crea la ventana del menú principal y muestra opciones para abrir la ventana de "Objetos perdidos" y consultar los objetos perdidos existentes.

La aplicación utiliza una base de datos MySQL para almacenar los objetos perdidos. Los parámetros de conexión, como el host, usuario, contraseña y nombre de la base de datos, se definen en cada función que interactúa con la base de datos.

# Código:

import tkinter as tk

from tkinter import ttk

import mysql.connector

from tkinter import messagebox

def center\_window(window):

    window.update\_idletasks()

    width = window.winfo\_width()

    height = window.winfo\_height()

    x = (window.winfo\_screenwidth() // 2) - (width // 2)

    y = (window.winfo\_screenheight() // 2) - (height // 2)

    window.geometry(f"{width}x{height}+{x}+{y}")

def agregar\_objeto\_perdido(nombre, categoria, descripcion):

    # Conectar a la base de datos

    conexion = mysql.connector.connect(

        host="localhost",

        user="root",

        password="Isai1622?",

        database="ñññ"

    )

    cursor = conexion.cursor()

    # Insertar los valores en la base de datos

    consulta = "INSERT INTO objetos\_perdidos (nombre, categoria, descripcion) VALUES (%s, %s, %s)"

    valores = (nombre, categoria, descripcion)

    cursor.execute(consulta, valores)

    conexion.commit()

    # Cerrar la conexión

    cursor.close()

    conexion.close()

def eliminar\_objeto\_perdido(id\_objeto):

    # Conectar a la base de datos

    conexion = mysql.connector.connect(

        host="localhost",

        user="root",

        password="Isai1622?",

        database="ñññ"

    )

    cursor = conexion.cursor()

    # Eliminar el objeto de la base de datos

    consulta = "DELETE FROM objetos\_perdidos WHERE id = %s"

    valor = (id\_objeto,)

    cursor.execute(consulta, valor)

    conexion.commit()

    # Cerrar la conexión

    cursor.close()

    conexion.close()

def consultar\_objetos\_perdidos():

    # Conectar a la base de datos

    conexion = mysql.connector.connect(

        host="localhost",

        user="root",

        password="Isai1622?",

        database="ñññ"

    )

    cursor = conexion.cursor()

    # Realizar la consulta SELECT

    consulta = "SELECT \* FROM objetos\_perdidos"

    cursor.execute(consulta)

    resultados = cursor.fetchall()

    # Crear una nueva ventana para mostrar los resultados

    ventana\_resultados = tk.Toplevel()

    ventana\_resultados.title("Resultados de la consulta")

    ventana\_resultados.configure(bg="#F0F0F0")

    ventana\_resultados.geometry("600x400")

    ventana\_resultados.resizable(False, False)

    # Centrar la ventana de resultados

    center\_window(ventana\_resultados)

    # Crear el widget Treeview para mostrar los resultados en una tabla

    treeview = ttk.Treeview(ventana\_resultados)

    treeview["columns"] = ("id", "nombre", "categoria", "descripcion")

    # Establecer los encabezados de la tabla

    treeview.heading("id", text="ID", anchor=tk.W)

    treeview.heading("nombre", text="Nombre", anchor=tk.W)

    treeview.heading("categoria", text="Categoría", anchor=tk.W)

    treeview.heading("descripcion", text="Descripción", anchor=tk.W)

    # Configurar el tamaño de las columnas

    treeview.column("id", width=50, anchor=tk.W)

    treeview.column("nombre", width=100, anchor=tk.W)

    treeview.column("categoria", width=100, anchor=tk.W)

    treeview.column("descripcion", width=350, anchor=tk.W)

    # Alinear todas las columnas a la izquierda

    for column in treeview["columns"]:

        treeview.column(column, anchor=tk.W)

    # Agregar los resultados a la tabla

    for resultado in resultados:

        treeview.insert("", "end", values=resultado)

    # Mostrar la tabla en la ventana

    treeview.pack(fill="both", expand=True)

    def eliminar\_objeto():

        # Obtener el ID del objeto seleccionado

        seleccion = treeview.focus()

        if seleccion:

            id\_objeto = treeview.item(seleccion)["values"][0]

            # Mostrar un mensaje de confirmación antes de eliminar el objeto

            confirmacion = tk.messagebox.askyesno("Confirmar eliminación", "¿Estás seguro de eliminar este objeto?")

            if confirmacion:

                # Eliminar el objeto de la base de datos

                eliminar\_objeto\_perdido(id\_objeto)

                # Eliminar el objeto de la tabla

                treeview.delete(seleccion)

    # Botón para eliminar un objeto

    eliminar\_btn = ttk.Button(ventana\_resultados, text="Eliminar", command=eliminar\_objeto)

    eliminar\_btn.pack(pady=10)

    # Cerrar la conexión

    cursor.close()

    conexion.close()

def abrir\_ventana\_objetos\_perdidos():

    global ventana\_objetos\_perdidos  # Declarar la variable como global

    # Cerrar la ventana del menú principal

    root.withdraw()

    # Crear una nueva ventana para "Objetos perdidos"

    ventana\_objetos\_perdidos = tk.Toplevel()

    ventana\_objetos\_perdidos.title("Objetos perdidos")

    ventana\_objetos\_perdidos.configure(bg="#F0F0F0")

    ventana\_objetos\_perdidos.geometry("400x400")

    ventana\_objetos\_perdidos.resizable(False, False)

    # Centrar la ventana de "Objetos perdidos"

    center\_window(ventana\_objetos\_perdidos)

    # Etiqueta centrada con el título "Objetos perdidos"

    titulo\_label = tk.Label(ventana\_objetos\_perdidos, text="Objetos perdidos", font=("Arial", 16))

    titulo\_label.pack(pady=10)

    # Campos para agregar objetos perdidos

    nombre\_label = ttk.Label(ventana\_objetos\_perdidos, text="Nombre:")

    nombre\_label.pack()

    nombre\_entry = ttk.Entry(ventana\_objetos\_perdidos)

    nombre\_entry.pack()

    categoria\_label = ttk.Label(ventana\_objetos\_perdidos, text="Categoría:")

    categoria\_label.pack()

    categoria\_entry = ttk.Entry(ventana\_objetos\_perdidos)

    categoria\_entry.pack()

    descripcion\_label = ttk.Label(ventana\_objetos\_perdidos, text="Descripción:")

    descripcion\_label.pack()

    descripcion\_entry = ttk.Entry(ventana\_objetos\_perdidos)

    descripcion\_entry.pack()

    # Botón para agregar objetos perdidos

    agregar\_btn = ttk.Button(ventana\_objetos\_perdidos, text="Agregar",

                             command=lambda: agregar\_objeto\_perdido(nombre\_entry.get(), categoria\_entry.get(), descripcion\_entry.get()))

    agregar\_btn.pack(pady=10)

    # Botón para regresar al menú principal

    regresar\_btn = ttk.Button(ventana\_objetos\_perdidos, text="Regresar al Menú Principal", command=regresar\_al\_menu)

    regresar\_btn.pack(pady=10)

def regresar\_al\_menu():

    # Cerrar la ventana de "Objetos perdidos"

    ventana\_objetos\_perdidos.destroy()

    # Mostrar la ventana del menú principal

    root.deiconify()

def main\_menu():

    global root

    root = tk.Tk()

    root.title("Menú principal")

    root.configure(bg="#F0F0F0")

    root.geometry("400x400")

    root.resizable(False, False)

    # Centrar la ventana principal

    center\_window(root)

    # Etiqueta centrada con el título "Menú principal"

    titulo\_label = tk.Label(root, text="Menú principal", font=("Arial", 16))

    titulo\_label.pack(pady=10)

    # Botón para abrir la ventana de "Objetos perdidos"

    abrir\_ventana\_btn = ttk.Button(root, text="Ir a Objetos perdidos", command=abrir\_ventana\_objetos\_perdidos)

    abrir\_ventana\_btn.pack(pady=10)

    # Botón para consultar objetos perdidos

    consultar\_btn = ttk.Button(root, text="Consultar", command=consultar\_objetos\_perdidos)

    consultar\_btn.pack(pady=10)

    root.mainloop()

# Ejecutar el menú principal

main\_menu()

# Funcionamiento

Se despliega un menú principal que incluye el botón de consultar y el de Objetos perdidos:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Al pulsar el botón de objetos perdidos nos despliega una ventana para agregar un objeto perdido a la base de datos mysql.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Para ver el objeto agregado, hay que ir a la ventana de consultas y así visualizarlo:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Para eliminar el objeto hay que seleccionar el objeto y pulsar el botón eliminar, nos aparecerá una ventana de confirmación:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Confirmamos y el objeto se elimina.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente