UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

LENGUAJES FORMALES Y DE PROGRAMACIÓN

PRIMER SEMESTRE 2024

Manual Técnico

Descripción General

El código proporcionado es un programa en Python que utiliza la biblioteca Tkinter para crear una interfaz gráfica de usuario (GUI). El programa permite cargar datos de estudiantes desde un archivo de texto, cargar calificaciones de los estudiantes desde otro archivo de texto, generar informes en formato HTML sobre los datos de los estudiantes y generar un informe de aprobación de estudiantes basado en sus calificaciones. Además, proporciona la funcionalidad para mostrar los tres mejores estudiantes con el promedio más alto.

Requisitos del Sistema

- Python 3.x instalado en el sistema.
- La biblioteca Tkinter incluida en la instalación de Python.

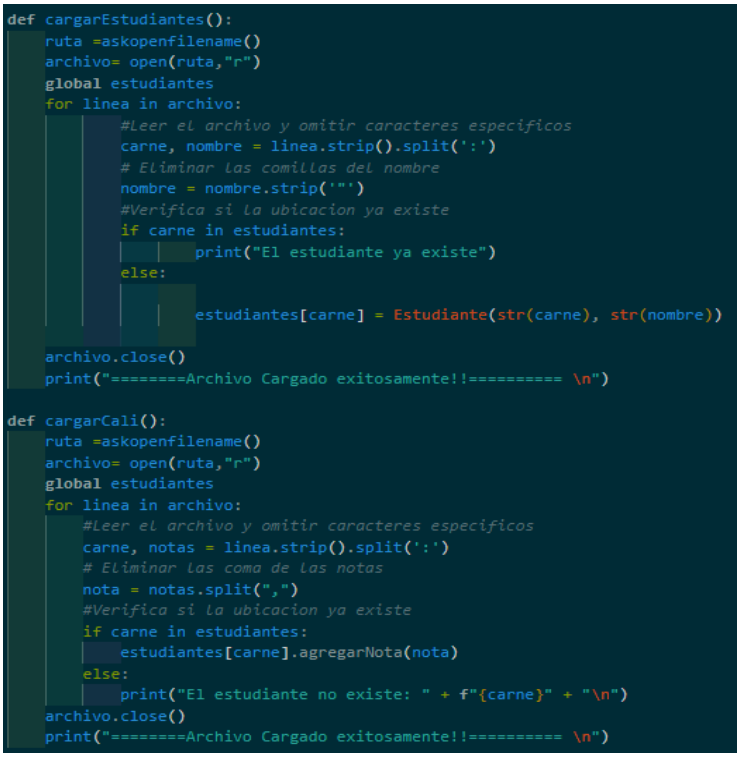
Instalación y Configuración

1. Asegúrese de tener Python 3.x instalado en su sistema.
2. No se requiere una configuración adicional para la biblioteca Tkinter, ya que es una biblioteca estándar de Python.
3. Asegúrese de tener los archivos de datos necesarios en el mismo directorio que el script principal, o modifique las rutas de archivo según sea necesario.

Estructura del Código

El código consta de las siguientes partes:

1. **Importación de Módulos**: Se importan los módulos necesarios, incluyendo `askopenfilename` de la biblioteca Tkinter y un módulo personalizado `Estudiante`.
2. **Funciones Principales**:
    - `cargarEstudiantes()`: Carga datos de estudiantes desde un archivo de texto y los almacena en un diccionario `estudiantes`.
    - `cargarCali()`: Carga calificaciones de estudiantes desde un archivo de texto y las asigna a los estudiantes correspondientes en el diccionario `estudiantes`.
    - `informeInventario()`: Genera un informe HTML con los datos de los estudiantes.
    - `reporteAprobacion()`: Genera un informe HTML que muestra el estado de aprobación de los estudiantes.
    - `top3Estudiantes()`: Genera un informe HTML que muestra los tres mejores estudiantes con el promedio más alto.
    - `bubbleSort()`: Implementa el algoritmo de ordenamiento de burbuja para ordenar los estudiantes según su promedio.



```
def cargarEstudiantes():
    ruta =askopenfilename()
    archivo= open(ruta,"r")
    global estudiantes
    for linea in archivo:
            #Leer el archivo y omitir caracteres especificos
            carne, nombre = linea.strip().split(':')
            # Eliminar las comillas del nombre
            nombre = nombre.strip('"')
            #Verifica si la ubicacion ya existe
            if carne in estudiantes:
                print("El estudiante ya existe")
            else:

                estudiantes[carne] = Estudiante(str(carne), str(nombre))

    archivo.close()
    print("========Archivo Cargado exitosamente!!========== \n")

def cargarCali():
    ruta =askopenfilename()
    archivo= open(ruta,"r")
    global estudiantes
    for linea in archivo:
        #Leer el archivo y omitir caracteres especificos
        carne, notas = linea.strip().split(':')
        # Eliminar las coma de las notas
        nota = notas.split(",")
        #Verifica si la ubicacion ya existe
        if carne in estudiantes:
            estudiantes[carne].agregarNota(nota)
        else:
            print("El estudiante no existe: " + f"{carne}" + "\n")
    archivo.close()
    print("========Archivo Cargado exitosamente!!========== \n")
```

3. **Funciones Auxiliares**:
    - `mostrar_menu()`: Muestra un menú de opciones para el usuario.

```python
def mostrar_menu():
    print("1. Cargar Estudiantes")
    print("2. Cargar Calificaciones")
    print("3. Crear Informe de General de Estudiantes")
    print("4. Reporte de aprobacion de estudiantes")
    print("5. Top 3 estudiantes con mejor promedio")
    print("6. Salir")
    print("")

#Hola aux
while True:
    mostrar_menu()
    opcion = input("Elige una opción: ")
    if opcion == "1":
        print("===== Cargando Estudiantes.... ========")
        print("")
        cargarEstudiantes()
    elif opcion == "2":
        print("========= Cargando Calificaciones..... ===========")
        print("")
        cargarCali()
    elif opcion == "3":
        print("Creando informe, espere...." + "\n")
        informeInventario()
        print("Revise el archivo .html" + "\n")
    elif opcion == "4":
        print("Creando informe, espere...." + "\n")
        reporteAprobacion()
        print("Revise el archivo .html" + "\n")
    elif opcion == "5":
        print("Creando informe, espere...." + "\n")
        top3Estudiantes()
        print("Revise el archivo .html" + "\n")
    elif opcion == "6":
        print("Saliendo..." + "\n")
        break
    else:
        print("Opción inválida. Por favor, elige una opción del menú." + "\n")

#Todo los derechos reservados 2024©
```

4. **Bucle Principal**: Un bucle `while` que muestra el menú y permite al usuario elegir las opciones.

Uso del Programa

- Ejecute el script en un entorno de Python.
- Elija las opciones del menú según sea necesario para cargar datos de estudiantes, cargar calificaciones, generar informes o salir del programa.
- Revise los archivos .html generados para obtener los informes.

Consideraciones de Desarrollo

- Asegúrese de tener los archivos de datos correctamente formateados según las expectativas del programa.
- El código utiliza un enfoque simple para generar informes HTML. Para aplicaciones más avanzadas, se pueden considerar bibliotecas Python dedicadas para la generación de HTML, como Jinja2 o Flask.

Este manual técnico proporciona una visión general del funcionamiento y la estructura del programa, así como instrucciones básicas para su uso. Para un análisis más detallado del código y su funcionamiento interno, consulte los comentarios en el propio script.