Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Lenguajes Formales y de Programación

Sección -B

MANUAL TÉCNICO

Kevin Gerardo Ruíz Yupe
201903791
19/08/2021

Objetivos

General:

Enfatizar en el uso de clases, variables y métodos escritos en la aplicación para usuarios con conocimiento específico en el tema. Asimismo, para edición de código para futuras actualizaciones

Específicos:

- Mostrar el IDE utilizado para el desarrollo del programa
- Describir el proceso de solución, así como las herramientas utilizadas durante el proyecto.

Introducción

Este manual pretende explicar y describir los temas que se desarrollan durante el proceso de creación del programa. De la misma forma, se indica los requerimientos del sistema, el IDE utilizado y las librerías que se importaron.

La principal función de esta aplicación es la obtención de datos a partir de un archivo con extensión .lfp, recorriendo cada carácter y haciendo uso de métodos propios de Python.
Se usa también algoritmos de ordenamientos para casos específicos en los reportes.

Descripción de la solución

La aplicación es una Clase de nombre ***Principal***, donde el método ***Menú*** genera la parte gráfica en la consola.

```
class Principal:
    def __init__(self):
```

En el constructor de nuestra clase declaramos las listas a utilizar

```
        self.Promedio_Estudiantes=0
        self.TituloCurso=''
        self.ListaEstudiantes=[]
        self.ListaNotas=[]
        self.ListaAprobados=[]
        self.NotasReprobados=[]
        self.ListaReprobados=[]
        self.NotasAprobados=[]
        self.Notas_Maximas=[]
        self.Nota_Maxima=0
        self.Notas_Minimas=[]
        self.Nota_Minima=0
        self.ListaOrdenadaNotas=[]
        self.ListaNombresOrdenados=[]
        self.ListaReportes=[]
```

Se muestra el método para el menú principal de la aplicación:

```
        print('|        Control de Notas       | \n'
        '|-------------------------------|\n'
        '| 1. Cargar Archivo             | \n'
        '| 2. Mostrar reporte en consola |\n'
        '| 3. Exportar reporte           |\n'
        '| 4. Salir                      |\n'
        '|_______________________________|')
        print('\n')
        print('Seleccione una opción: ')
        Opcion=int(input('> '))
        if(Opcion==1):
            self.abrir2=self.Abrir()
            if self.abrir2==None:
                print('No se eligió archivo. Intente de nuevo')
                self.MostrarConsola()
            else:
                print('---------- Archivo de texto -------')
                print(self.abrir2)
                print("-----------------------------------")
                self.MostrarConsola()
        elif(Opcion==2):
            self.QuitarSaltos(self.abrir2)
```

Se importaron distintas librerías para la parte de la interfaz gráfica.

En este método se crea una variable para abrir el archivo, se verifica que nuestra variable no sea vacía o nula. Si está vacía, el método retorna valor vacío igualmente. Si no, se crea otra variable para la lectura, y el método retorna el contenido del archivo.

```
def Abrir(self):
    #Tk().withdraw()
    ArchivoLFP=filedialog.askopenfile(initialdir='./',title='Seleccione una archivo .LFP',filetypes=(('Archivos .lfp','
    if ArchivoLFP is None:
        return None
    else:
        self.TextoLFP=ArchivoLFP.read()
        ArchivoLFP.close()
        return self.TextoLFP
```

Cuando procesamos todo el texto del archivo se guardará en listas auxiliares para verificar espacios o signos que no deseemos.

```
            self.ReportesComas=self.ReportesComas+elementos
        self.ReportesComas2=self.ReportesComas+','
        self.reportessincomas=[]
        self.r=''
        self.r2=''
        for x in self.ReportesComas2:
            if x!=',':
                self.r=self.r+x
            else:
                self.r2=self.r
                self.reportessincomas.append(self.r)
                self.r=''

        self.NotasEstudiantes=[]
        print('Curso: '+ self.titulo)
        self.TituloCurso=self.titulo.replace('_',' ')
        for i in self.NotasEstudiante:
            self.NotasEstudiantes.append(round(float(i)))
```

Recorremos la lista de reportes pedidos y verificamos qué reportes son los que se operarán

```
for i in self.reportessincomas:
    if i=='DESC':
        self.OrdenamientoDESC(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='MAX':
        self.NotasMaximas(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='ASC':
        self.OrdenamientoASC(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='APR':
        self.Aprobados(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='REP':
        self.Reprobados(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='MIN':
        self.NotasMinimas(self.NotasEstudiantes,self.NombreEstudiantes)
    elif i=='AVG':
        self.Promedio(self.NotasEstudiantes)
```

Observamos que cada validación tiene como valor verdadero llamar a otro método que ejecutará la función que indica su nombre. Ejemplo:

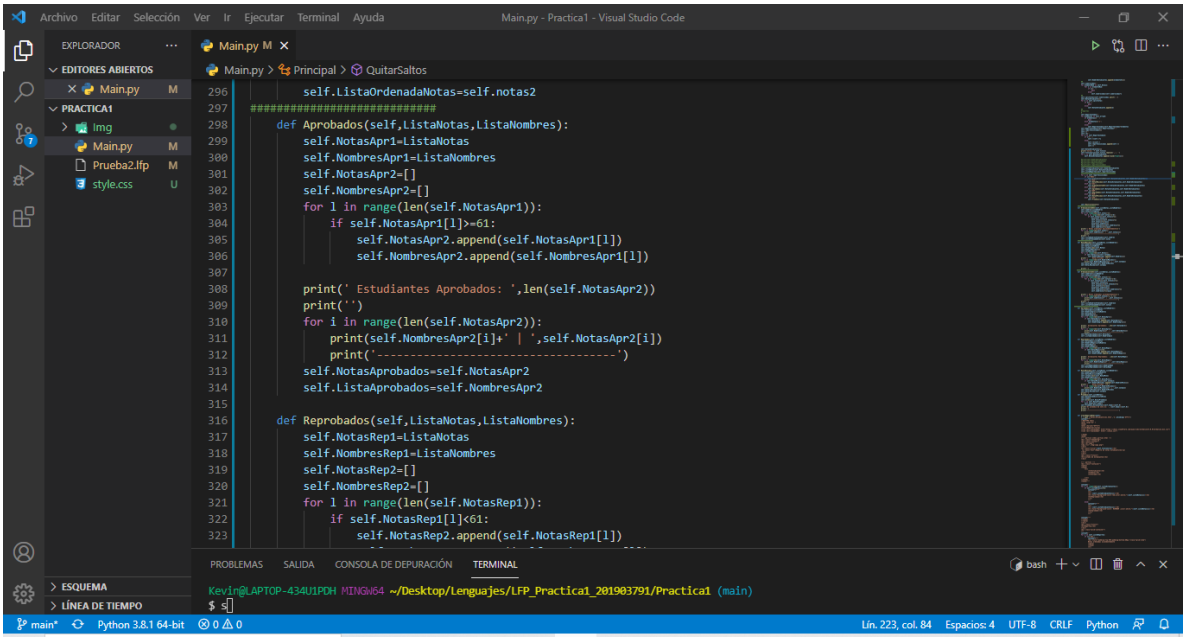
Este método, como ejemplo, realiza el proceso para verificar los alumnos aprobados, es decir, con una mayor o igual a 61. Se crean los demás métodos cada uno con su función principal.

```
def Aprobados(self,ListaNotas,ListaNombres):
    self.NotasApr1=ListaNotas
    self.NombresApr1=ListaNombres
    self.NotasApr2=[]
    self.NombresApr2=[]
    for l in range(len(self.NotasApr1)):
        if self.NotasApr1[l]>=61:
            self.NotasApr2.append(self.NotasApr1[l])
            self.NombresApr2.append(self.NombresApr1[l])

    print(' Estudiantes Aprobados: ',len(self.NotasApr2))
    print('')
```

Editor de código utilizado para la aplicación

Visual Studio Code de Microsoft



Requerimientos para Python

- Modern Operating System:
  - Windows 7 or 10
  - Mac OS X 10.11 or higher, 64-bit
  - Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

Librerías utilizadas

```
from tkinter import filedialog,Tk
from tkinter.constants import NONE
from typing import List, Text
```