

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

Lenguajes Formales y de Programación  
Sección "B-"



## **“MANUAL TÉCNICO”**

Samuel Alejandro Pajoc Raymundo

Carné: 201800665

# Objetivos

## General:

Proporcionar al lector una explicación más detallada de algunas clases especiales o estructuras establecidas, así como brindar una explicación sobre la utilización de atributos, métodos y del desarrollo de las distintas funciones que permiten que la aplicación opere correctamente, así como para poder implementar modificaciones que actualicen el buen funcionamiento.

## Específicos:

- Mostrar una descripción detallada de las funciones y métodos principales de la aplicación, así como la estructura que debe cumplir el archivo de entrada.
- Proporcionar al lector una explicación técnica o formal de los distintos procesos que la conforman.

# Introducción

Este manual contiene una explicación más detallada del funcionamiento de la aplicación, el motivo de este manual es poder servir de guía para poder realizar cambios en los distintos procesos si estos resultan ser obsoletos, por lo que, se motiva al lector a poder realizar una pequeña implementación en la simplificación de cualquiera de los distintos procesos que conforman a esta aplicación.

La aplicación tiene como objetivo leer, extraer y procesar los distintos datos de un grupo de estudiantes pertenecientes a cierto curso, todo esto a partir de un archivo de texto plano con formato lfp. Con los datos se obtenidos se realizan distintos ordenamientos solicitados por las instrucciones contenidas en el archivo de entrada, para luego ser analizados por el usuario.

# Descripción de la Solución

Antes de iniciar con la explicación de los distintos procesos de la aplicación, se explicará la estructura con la que debe cumplir el archivo de entrada, por lo que si esto no se cumple, no se garantiza un buen funcionamiento de la misma.

- En la primer línea del archivo debe contener el “Nombre del curso”, a este identificador le prosigue un signo igual ( = ) y un par de llaves( { } ), las cuales contendrán el listado de estudiantes pertenecientes a dicho curso junto a su respectiva nota.
- El listado de estudiantes se compone por elementos separados por coma ( , ).
- Cada estudiante iniciará con un signo menor que ( < ), luego sigue el nombre del alumno, el cual estará dentro de comillas dobles( “ ” ), a continuación se encuentra un punto y coma ( ; ), luego continua con la nota obtenida en tal curso, por ultimo finaliza con un signo mayor que( > ), seguido de una coma, la cual indicará que a continuación sigue otro alumno.

Los distintos parámetros que permiten ordenar las notas son solicitados con palabras reservadas, las cuales son las siguientes:

- ASC = Ordenamiento Ascendente.
- DESC = Ordenamiento Descendente.
- AVG = Obtener Promedio (de todas las notas de un mismo curso).
- MIN = Obtener Nota Mínima (la nota más pequeña).
- MAX = Obtener Nota Máxima (la nota más grande).
- APR = Obtener Cantidad de estudiantes aprobados.
- REP = Obtener Cantidad de estudiantes reprobados.

Entre las consideraciones encontramos con mayor prioridad están:

- No es necesario que cada estudiante contenga separación con espacios en blanco o que únicamente se encuentre un estudiante por línea.
- Separar a cada estudiante por coma o este no será tomado en cuenta y será indicio de error, a menos que sea el ultimo estudiante, este no importa si contiene o no

coma, ya que se desarrolló la aplicación de tal forma que detecte este caso y funcione con normalidad.

- Al escoger una opción en el menú principal, siempre ingresar una letra, de lo contrario se mostrará una alerta indicando que únicamente se debe ingresar números enteros, y únicamente los que se especifica en el menú, debido a que si se ingresa otro número distinto a los indicados, el menú seguirá mostrándose junto al mensaje de advertencia de solo números.
- Si no se elige un archivo de entrada no se podrá ingresar a la opción dos ni a la opción tres, y si se selecciona un archivo de entrada, pero no se ingresa a la opción dos, no se podrá ingresar a la opción tres, por lo que se recomienda siempre seguir el orden de las opciones.

# Funcionamiento

Para evitar fallos con los datos de entrada en el menú, se utilizó un try-except (posee la misma utilidad que un try-catch), tomando en cuenta todos los errores posibles con ValueError.

Después de seleccionar un archivo, se verifica que en cada línea dentro de las llaves contenga únicamente un estudiante por línea por lo menos, si en dado caso existen más, estos serán separados por medio de un Split, tomando como separador las comas contenidas en las líneas, esto se realiza por medio de un conteo de comas por línea, si en una misma línea existen más de una coma, entonces se realizará esta separación, de lo contrario se reiniciará el conteo de comas por línea.

La forma en que se analiza cada línea del archivo de entrada es por medio del análisis de carácter por carácter, esto quiere decir que se arma un arreglo el cual contendrá una línea formada por cada carácter permitido o aceptado para el análisis posterior. De este modo se eliminan ciertos caracteres innecesarios, como lo son las llaves, si en dado caso se analizara un archivo sin llaves, este no afectará el buen funcionamiento de esta aplicación.

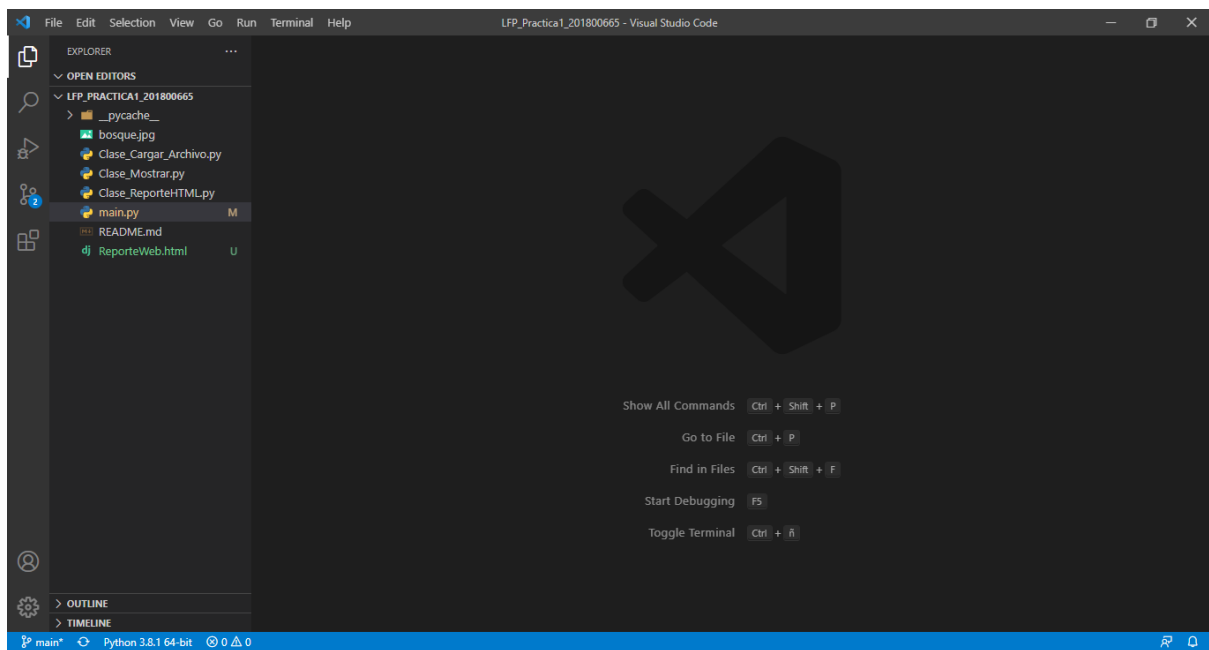
Si las palabras reservadas se encuentran más de una vez, esta se realizará una única vez y las demás serán desechadas, esto se realiza para tener la mejor optimización de recursos de la computadora.

Si se desea mejorar la apariencia del reporte generado en la pagina web, es necesario recordar que debe contener conocimientos de HTML, ya que las funciones que permiten el desarrollo de esta, es una mezcla entre los lenguajes Python y HTML.

Si posee alguna inquietud con el funcionamiento de la aplicación, no dude consultar el código fuente, ya que este posee ciertos comentarios añadidos por el desarrollador, en las áreas en las cuales pueden surgir más dudas en su funcionamiento.

# Editor de Código

El editor con que se creó la aplicación fue Visual Studio Code, ya que se trabajó bajo el lenguaje Python, se optó por esta herramienta, ya que funciona de forma muy optima con el lenguaje, además de brindar ciertas herramientas de gran utilidad, como lo es el versionamiento con GitHub, por lo que realizar los realices fue mucho más sencillo con este editor. Por lo que se observó que la utilización de este editor fue de gran beneficio para el programador, ya que sus ejecuciones fueron rápidas y eficientes.



## Requerimientos:

- Procesador de 1,6 GHz o superior.
- 1 GB de RAM o 1,5 si se ejecuta en una máquina virtual.
- 10 MB de espacio disponible en disco duro.
- Disco duro de 544 RPM
- Tarjeta de video compatible con DirectX 9 con una resolución de pantalla de 1024 X 768 o superior.
- Sistema Operativo:  
El sistema operativo en el que se llevó a cabo la practica fue Windows 10 de 64 bits, pero puede funcionar o desarrollarse en Windows 7.

# Librerías Utilizadas

Las librerías utilizadas para el desarrollo de este proyecto fueron:

```
from io import open
from tkinter import *
from tkinter import filedialog
import time
import webbrowser as wb
```