



MANUAL TÉCNICO

PRÁCTICA #1



Lenguajes Formales y de Programación

Javier Alejandro Gutierrez de León
202004765



Guatemala, 19 de agosto de 2021

Introducción

Se trata de una aplicación de consola, que permita la lectura de un archivo de texto plano con extensión “lfp” que contiene los datos de los estudiantes de un curso y la nota final que obtuvieron en dicho curso, finalmente el archivo contiene una palabra reservada que indica que tipo de reporte se quiere realizar, utilizando esta información para crear tablas de las notas en consola y en un archivo “html” el cual se puede visualizar en un explorador. Teniendo los siguientes apartados:

1. Cargar archivo: Muestra una ventana emergente que permite al usuario seleccionar un archivo lfp y carga el archivo a memoria, lo analiza y ejecuta el reporte.
2. Mostrar reporte en consola: Muestra los datos obtenidos durante la ejecución del reporte. Se debe mostrar en pantalla el nombre del curso, el total de estudiantes que contiene el archivo y los datos propios del reporte que se solicite.
3. Exportar reporte: Se debe de escribir un archivo HTML con los datos del reporte generado, la manera en que los datos son mostrados deben de ser agradables al usuario. Por lo menos, debe mostrar el nombre del curso, en una tabla la lista de estudiantes y su respectiva nota (en rojo las notas reprobadas y en azul las aprobadas) y al final los parámetros solicitados.
4. Salir: Termina la ejecución de la aplicación.

Objetivos

Objetivos Generales

- ✓ Aplicar los conceptos adquiridos en el curso de Lenguajes Formales y de Programación en el desarrollo de la aplicación.

Objetivos Específicos

- ✓ Aplique los conceptos generales sobre lenguajes formales, tales como alfabeto, símbolos, cadenas y reglas.
- ✓ Conozca las características principales del lenguaje de programación Python.
- ✓ Aplique algoritmos de ordenamiento.

Especificación Técnica

Requisitos de Hardware

- RAM: 128 MB
- Procesador: Mínimo Pentium 2 a 266 MHz

Requisitos de software

- Sistema operativo

Depende del buscador que se utilice

- Exploradores:

- | | |
|--|--|
| • Google Chrome (en todas las plataformas) | • Microsoft Edge (Windows, Android, iOS, Windows 10 móvil) |
| • Mozilla Firefox (en todas las plataformas) | • Safari (Mac, iOS) |
| • Internet Explorer (Windows) | • Opera (Mac, Windows) |

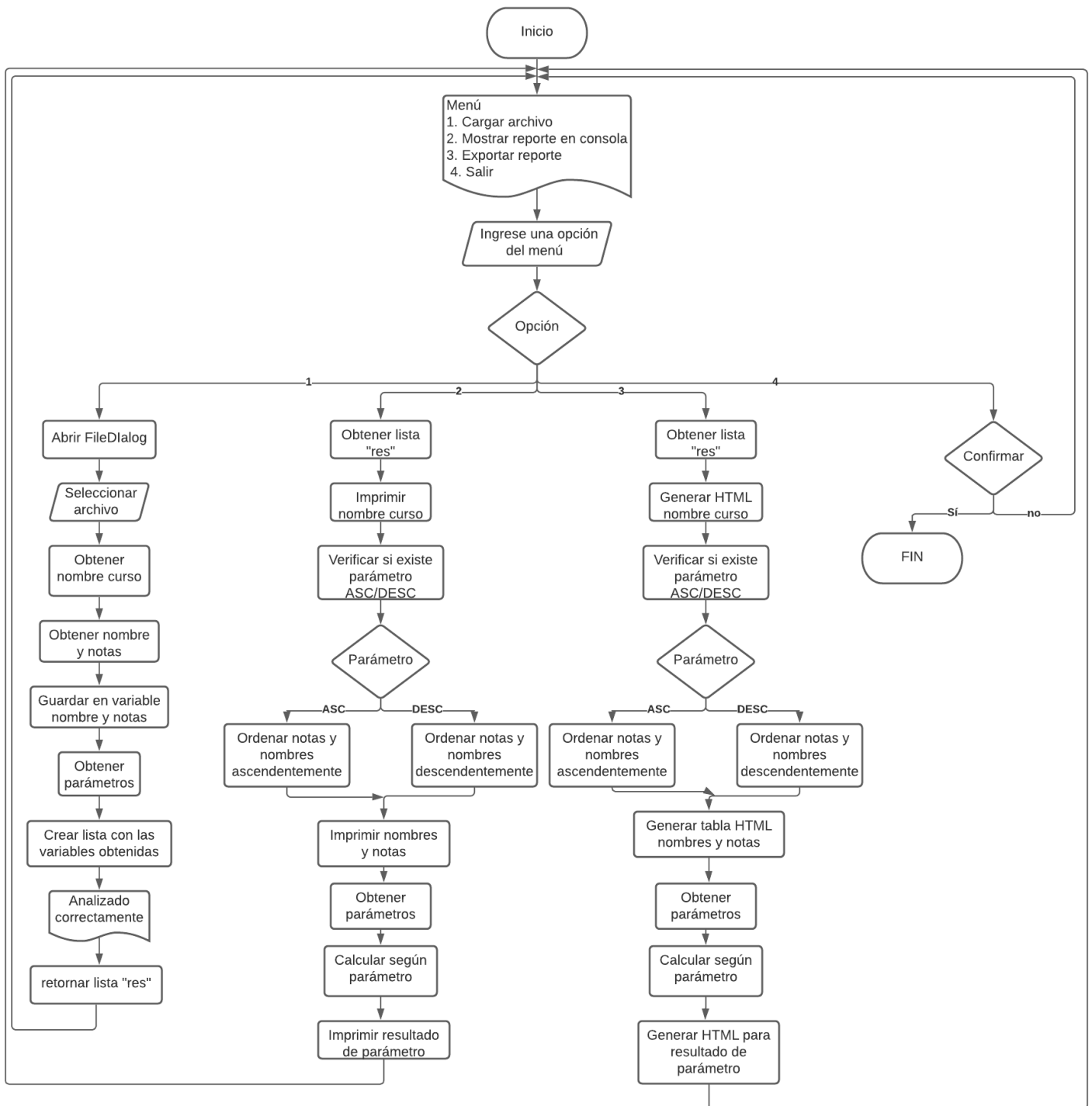
- Lenguaje de Programación e IDE

Para el desarrollo de este software se utilizó Python, y como entorno se utilizó Visual Studio Code

- Tecnologías utilizadas

- Python (v 3.8.10): Lenguaje de programación utilizado para desarrollar la aplicación.
- HTML5: lenguaje de marcado para la elaboración de páginas web, con el que se realizaron los reportes generados desde Python.
- Tkinter: Librería para utilizar FileDialog para buscar el archivo “*lfp*” que se quiere procesar.
- Bootstrap: Librería para el diseño del reporte HTML

DIAGRAMA DE FLUJO



Lógica del programa (Métodos principales)

Nombre	Descripción	Parámetros
LeerArchivo()	Método con el cual se busca el archivo "lfp" utilizando la librería Tkinter para luego leer línea por línea guardando el nombre del curso y los parámetros solicitados al final y guardando los nombres y notas de los alumnos en listas. Por último retorna la lista "res" la cual contiene en su primer espacio el nombre del curso, seguido de las listas de los datos de los alumnos y por último los parámetros solicitados, para utilizar esta información en los demás métodos.	<ul style="list-style-type: none"> • Ninguno
MostrarConsola(res)	Descompone la lista "res", obteniendo el nombre del curso luego imprimiendolo, obteniendo los parámetros e identificando el orden en que se deben ordenar los nombres, para proceder a obtener las listas de nombres y notas para imprimirlas en una tabla de ASCII con espacios definidos para mantener uniformidad en la tabla y por último imprime los demás datos solicitados por los parámetros, como promedio, mayor y menor nota y numero de estudiantes aprobados y reprobados.	<ul style="list-style-type: none"> • Lista "res" obtenida de la ejecución de "LeerArchivo()" (que el usuario ejecute la primera opción del menú) la cual contiene en su primer espacio el nombre del curso, seguido de las listas de los datos de los alumnos y por último los parámetros solicitados, para utilizar esta información en los demás métodos.
GenerarReporte(res)	Guarda en una variable el encabezado que lleva un archivo HTML, luego descompone la lista "res", obteniendo el nombre del curso luego guardandolo en una variable, guardando en una variable el código para generar una tabla con los datos en el orden en el que fueron ingresados en el archivo "lfp",	<ul style="list-style-type: none"> • Lista "res" obtenida de la ejecución de "LeerArchivo()" (que el usuario ejecute la primera opción del menú) la cual contiene en su primer espacio el nombre del curso, seguido de las listas de los datos de los alumnos y por último los parámetros solicitados, para utilizar esta

	<p>luego obteniendo los parámetros e identificando el orden en que se deben ordenar los nombres, para proceder a obtener las listas de nombres y notas generando un código para hacer una tabla para guardarlas en una variable y luego se guardan en variables los demás datos solicitados por los parámetros, como promedio, mayor y menor nota y número de estudiantes aprobados y reprobados. Por último se concatenan las variables en una sola la cual se manda escribir en un archivo HTML.</p>	<p>información en los demás métodos.</p>
Despedirse()	<p>Método con el cual se imprime el mensaje de despedida (ASCII art).</p>	<ul style="list-style-type: none"> • Ninguno

Conclusiones

El uso de la función `.split()` facilitó encontrar ciertos simbolos, para identificar cadenas que se encontraban dentro del formato establecido para los archivos “.lfp”.

El uso de Python para esta práctica facilitó el uso de muchas funciones que con otros lenguajes de programación son mas tardadas de desarrollar, reduciendo considerablemente el tiempo empleado para la codificación, además de ser mas amigable ya que a diferencia de otros lenguajes de programación se cometían menos errores.