

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y PROGRAMACIÓN



PRACTICA 1

JOAQUIN EMMANUEL ALDAIR COROMAC HUEZO
201903873
GUATEMALA, 17 DE FEBRERO DE 2021



MANUAL TÉCNICO PRACTICA 1

El manual muestra y describe las funciones del programa así como su lógica y paradigmas utilizados

Este manual contiene información sobre:

1. Lógica Programa
2. Paradigmas en Uso
3. Especificaciones
4. Control de versiones

Lógica Programa

➤ Archivo(main.py)

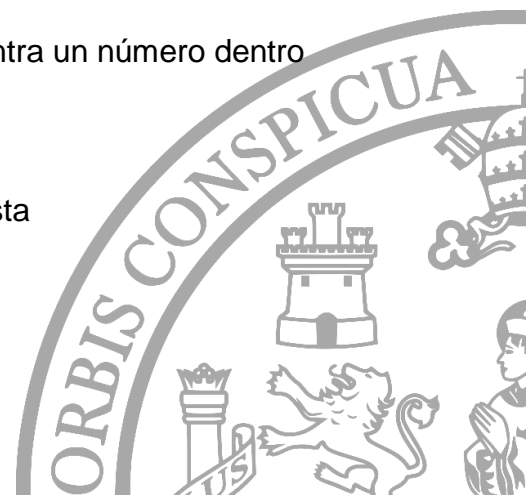
- Funciones
 - def main():
Función principal para ejecutar el programa y llama a la función Menu del archivo practica.py

➤ Archivo(practica.py)

- Funciones
 - def Menu():
Función donde el usuario realiza cada opción.

➤ Archivo(funciones.py)

- Funciones
 - def abrirArchivo():
Función mediante filedialog escoge el archivo de entrada.
 - def accion(linea,lista):
Función que divide cada línea del archivo en una lista.
 - def Ordenar(Lista):
Función que ordena la lista.
 - def Buscar(Lista, x):
Función que busca las posiciones en que se encuentra un número dentro de la lista.
 - def Ordenada(linea):
Función que imprime el resultado de ordenar una lista



- def Buscada(linea):

Función que imprime el resultado de Buscar en una lista.

- def TodasOrdenar(linea):

Función que imprime el resultado de Ordenar, Buscar en una lista.

- def TodasBuscar(linea):

Función que imprime el resultado de Ordenar, Buscar en una lista.

- def Todas(lista):

Función que imprime todos los resultados.

- def TodasArchivos(lista):

Función que devuelve todos los resultados.

- def archivo(lista):

Función que realiza el archivo HTML con todos los resultados.

➤ Archivo(Estilo.css)

- Arreglos
 - body
 - table
 - table
 - td, th

➤ Archivo(TodosArchivo.html)

- HTML donde se imprimen los resultados.



Funciones Buscar y Ordenar

Ordenar

Para el ordenar la lista de números se utilizo un algoritmo conocido como Ordenamiento Burbuja, dicha función recibe como parámetro una lista de los números a ordenar retorna la lista ya ordenada.

Mediante dos bucle for va recorriendo la lista y verifica que el numero en la posición sea mayor a la siguiente y cambia de posición mediante una posición temporal "temp".

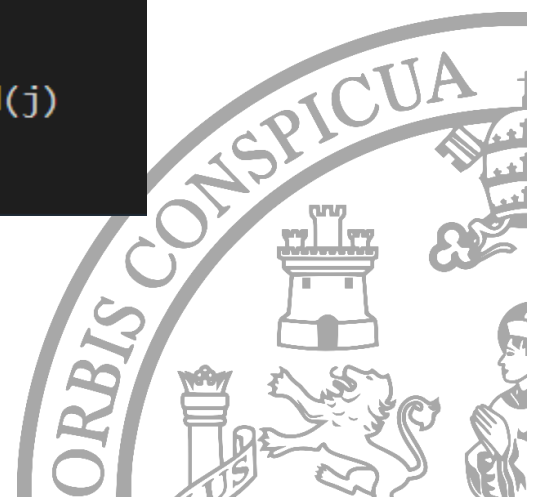
```
21
22 def Ordenar(Lista):
23     for no in range(len(Lista)-1,0,-1):
24         for i in range(no):
25             if Lista[i]>Lista[i+1]:
26                 temp = Lista[i]
27                 Lista[i] = Lista[i+1]
28                 Lista[i+1] = temp
29     return Lista
30
```

Buscar

Para buscar un número en la lista se utiliza la función Buscar que recibe como parámetro una lista y el número que se quiere encontrar.

Mediante un bucle for se recorre la lista hasta encontrar el número y toma la variable "i+1" que es la posición del número y lo agrega a una lista donde van las posiciones en que se encuentra el número y la cual retorna al finalizar el ciclo.

```
30
31 def Buscar(Lista, x):
32     posiciones=list()
33     for i in range(0,len(Lista)):
34         if Lista[i]==x:
35             j=str(i+1)
36             posiciones.append(j)
37     return posiciones
38
```



Paradigmas en Uso

Paradigma Imperativos

Estructurado

Este paradigma se orienta a mejorar la claridad, calidad y tiempo de desarrollo del programa de computadora recurriendo a estructuras básicas como secuencia, selección e iteración, para el proyecto se utilizaron estructuras de iteración (For & while) desde el menú hasta la lectura del archivo .txt.

Procedimental

Este paradigma consiste en basarse de un numero muy bajo de expresiones repetidas y englobarlas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse, para el proyecto se tiene un archivo Python dedicado a las funciones a utilizar en el programa.

Especificaciones

Especificaciones de Editor de Texto y versión Python

Visual Studio Code
python.3.8.1

Especificaciones de computadora

Procesador: Intel(R) Core™ i7-9750H CPU@
2.60 GHz 2.59GHz

RAM instalada: 16.0GB (15.9 GB usable)

Tipo de sistema: Sistema operativo de 64 bits,
Procesador basado en x64

Windows 10.



Control de versiones

Para el proyecto se trabajo con GitHub para controlar las versiones durante su realización.

Enlace Repositorio

https://github.com/jeach27/-LFP-Practica1_201903873.git

