

# Manual Técnico

Funciones principales del código:

```
def main():
```

Su principal función es mantener la ejecución del código persistente hasta que el usuario decida salir de la ejecución de la aplicación. Este ciclo muestra un menú repetidas veces para que el usuario pueda elegir que acción realizar dentro de la aplicación.

Cada parque se encarga del cargado de datos, de la lógica de mostrar los datos pertinentes de las acciones requeridas en el archivo cargado.

```
def open_file():
```

Su función principal es el parsing de datos. A través de una ventaja emergente para la selección del archivo del usuario. Una vez abierto este archivo, se separa por líneas y se parsea cada una individualmente. Se separa la línea en sus 3 componentes principales que son nombre, datos, comandos. Una vez estos datos son parseados se calculan los comandos debidos para cada una. Una vez hecho esto se ingresan los datos a

un array masivo que almacena los datos para cada línea. Este array contiene las listas ordenadas, los elementos buscados, los índices encontrados del elemento, las listas originales y sus nombres.

```
def open_file():
    file = fileopenbox("Python files", "Open file")
    print(f'Se selecciono el archivo {file}')
    #file = "C:\\Users\\Matus\\Documents\\USAC\\
    file1 = open(file, 'r')
    global Lines
    Lines = file1.readlines()

    for i in range(0, len(Lines)):
        Lines[i] = Lines[i].replace("\n", "")

    for line in Lines:
        # print("-----")
        # !Index of Equal sign
        equalI = line.index("=")
        letterI = len(line)
        # print("Equal Index:" + str(equalI))

        # !Index of First letter after data (refe
    open_file()
    for i in range(0, len(Lines))
```

```
def main():
    while True:
        print("-----")
        print("Bienvenido al programa de practica 1 de LFP")
        print("1. Cargar archivo de entrada")
        print("2. Desplegar listas ordenadas")
        print("3. Desplegar busquedas")
        print("4. Desplegar todas")
        print("5. Desplegar todas a archivo")
        print("6. Salir")
        print("-----")

        # open_file()#TODO debug only, delete

        option = input("Ingrese la opcion deseada \n")
        if (option == "1"):
            open_file()
            # print(f'megaArrays')
        elif (option == "2"):
            for result in megaArrays:
                if (result[5]):
                    print(result[0] + "ORDENADA -> " + str(result[2]))
        elif (option == "3"):
            for result in megaArrays:
                if (result[6] == True):
                    if (len(result[4])) > 1:
                        print(result[0] + " " + str(result[1]) + "---->DATO " + str(
                            result[3]) + " encontrado en indice(s) " + str(result[4]))
                    else:
                        print(result[0] + " " + str(result[1]) + "---->DATO " + str(
                            result[3]) + " no se encontro en ninguna posicion")
        elif (option == "4"):
            for result in megaArrays:
                if (result[5]):
                    print(result[0] + "ORDENADA -> " + str(result[2]))
                if (result[6] == True):
```

```
def save_html():
```

Esta función toma los elementos mas importantes para el usuario y los coloca en un sub-arreglo el cual va a ser convertido posteriormente a HTML. Adicional a esto se colocan los títulos de las respectivas columnas para una impresión mas fácil de leer para el usuario.

```
def tableToHTML(data):
```

Esta función toma el sub-array generado por `save_html()` y agrega el sintaxis adecuado para la generación correcta de una tabla HTML correcta y legible para cualquier navegador y parseador de DOM.

```
def orderArray(theArray):
```

Una simple función que toma un array dado, crea una copia (para evitar problemas de puntero) y la ordena con una implementación del algoritmo BubbleSort. Devuelve esta copia como resultado de la función.

```
def save_html():
    table = [{"Titulo", "Datos",
    for result in megaArrays:
        new = [result[0], result[
        if not result[5]: #5:Requ
            new[2] = empty_string
        if not result[6]:
            new[3] = empty_string
            new[4] = empty_string
        else:
            if len(new[4]) == 0:
                new[4] = "No enco
    table.append(new)
    for l in table:
        print(str(l))
    html_table = tableToHTML(tabl
    print(html_left + html_table

    with open('index.html', 'a')
        the_file.write(html_left

def tableToHTML(data):
    q = "<table>\n"
    for i in [(data[0:1], 'th'),
        q += "\n".join(
            [
```

```
def orderArray(theArray):
    n = len(theArray)
    arr = theArray.copy()

    for i in range(n):
        for j in range(0, n -
            if arr[j] > arr[j
                arr[j], arr[j
    return arr
```

## Manual de Usuario

Los elementos fundamentales para el funcionamiento de este programa son:

- Computadora compatible con Python 3.x
- Python 3.x
- Librería easygui 0.98.2 (pip install easygui)

Luego de la correcta instalación de los requisitos, se puede proceder a usar el programa.

- Menu principal:  
Este cuenta con un menú principal en el cual se muestran las opciones disponibles:

```

-----
Bienvenido al programa de practica 1 de LFP
1. Cargar archivo de entrada
2. Desplegar listas ordenadas
3. Desplegar busquedas
4. Desplegar todas
5. Desplegar todas a archivo
6. Salir
-----

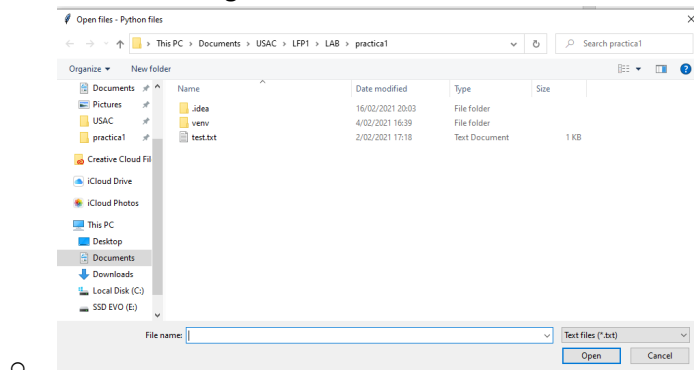
Ingrese la opcion deseada
|

```

- Carga de archivo:

Esta opción abre una ventana emergente de selección de archivo .txt el cual contiene los datos a cargar.

- Nota: La ventana emergente puede salir detrás del programa de ejecución y no enfrente, por lo que se sugiere minimizar el programa para poder acceder a esta ventana.
- Nota2: El programa asume que la sintaxis correcta de los datos fue empleada en el archivo de carga.

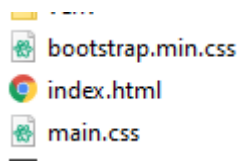


- Muestra de resultados (listas ordenadas/búsquedas/todas)

Estas funciones muestran respectivamente los datos procesados por el programa, de los datos cargados en el menú de carga.

- Desplegar todas a archivo

Esta opción genera un reporte del proceso de los datos en formato HTML el cual puede ser visualizado de mejor manera en una tabla.



TITULO	DATOS	ORDENADOS	BUSQUEDA	RESULTADOS BUSQUEDA
DATOS	[3, 4, 6, 5, 7, 1]	[1, 3, 4, 5, 6, 7]	---	---
LISTA1	[1, 4, 3, 5, 7, 5]	[1, 3, 4, 5, 5, 7]	5	[3, 5]
La	[3, 4, 5, 3, 1, 2]	---	7	No encontrado

