

# APL

Analizador y Procesador de Listas

**MANUAL TÉCNICO**

## Contenido

INTRODUCCIÓN: .....	3
REQUERIMIENTOS DEL SISTEMA .....	4
Windows: .....	4
Mac OS X: .....	4
Linux:.....	4
Menú .....	5
Cargar archivo de entrada: .....	6
Desplegar listas ordenadas: .....	7
Desplegar búsquedas: .....	8
Desplegar todas: .....	9
Desplegar todas a archivo.....	10
PARADIGMAS DE PROGRAMACIÓN UTILIZADAS .....	11
Programación estructurada.....	11
Programación procedimental .....	12
Programación modular .....	13
Programación Orientada a Objetos .....	13
CONCLUSIÓN: .....	14

## INTRODUCCIÓN:

Se realizó una aplicación, que permite analizar y procesar información de un archivo de texto dado con un formato, el cual contiene un identificador, lista de números y operaciones a realizar, se hizo a través del lenguaje Python, se hizo uso de HTML y CSS para generar reportes web y también se hizo uso de librerías externas como Tkinter para mostrar un explorador de archivos.

# REQUERIMIENTOS DEL SISTEMA

## Windows:

- Python 3.8.1
- Windows 10 (8u51 y superiores)
- Windows 8.x (escritorio)
- Windows 7 SP1 • Windows Vista SP2

## Mac OS X:

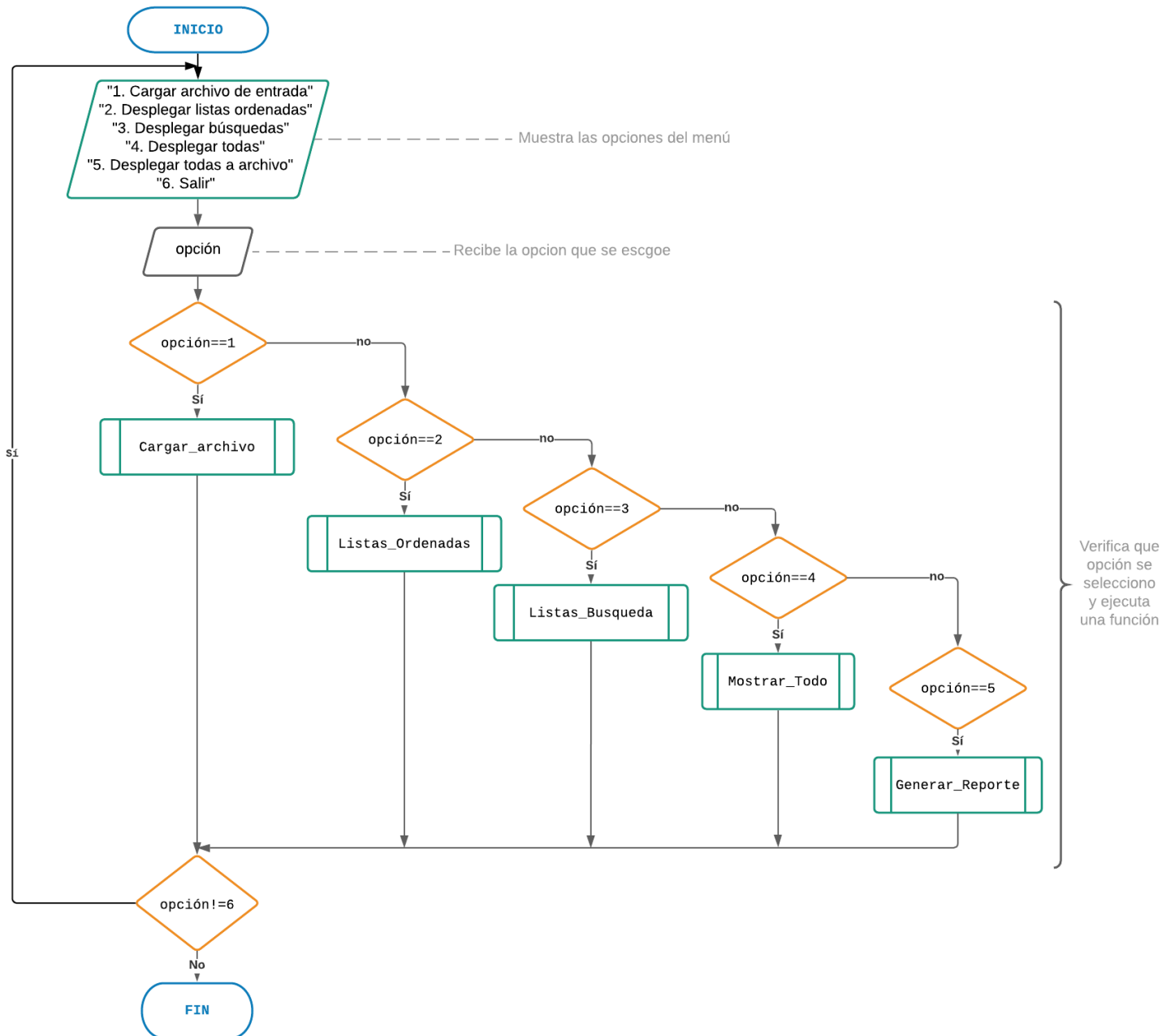
- Python 3.8.1
- Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- Privilegios de administrador para la instalación

## Linux:

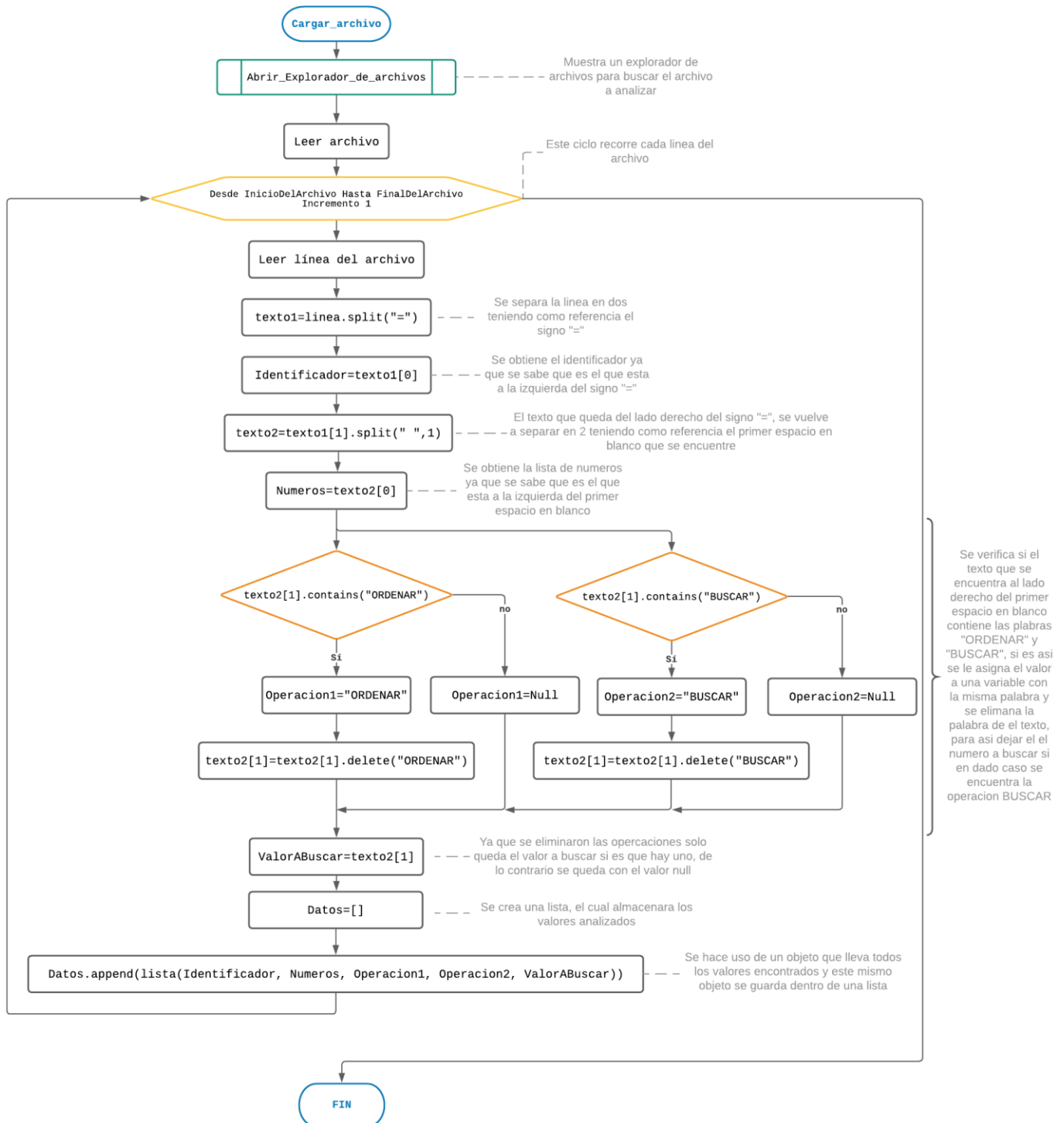
- Python 3.8.1
- Cualquier distribución de Linux.

En cualquiera de los sistemas que vaya a ejecutar la aplicación se debe contar con un navegador que soporte HTML5, para poder visualizar el reporte.

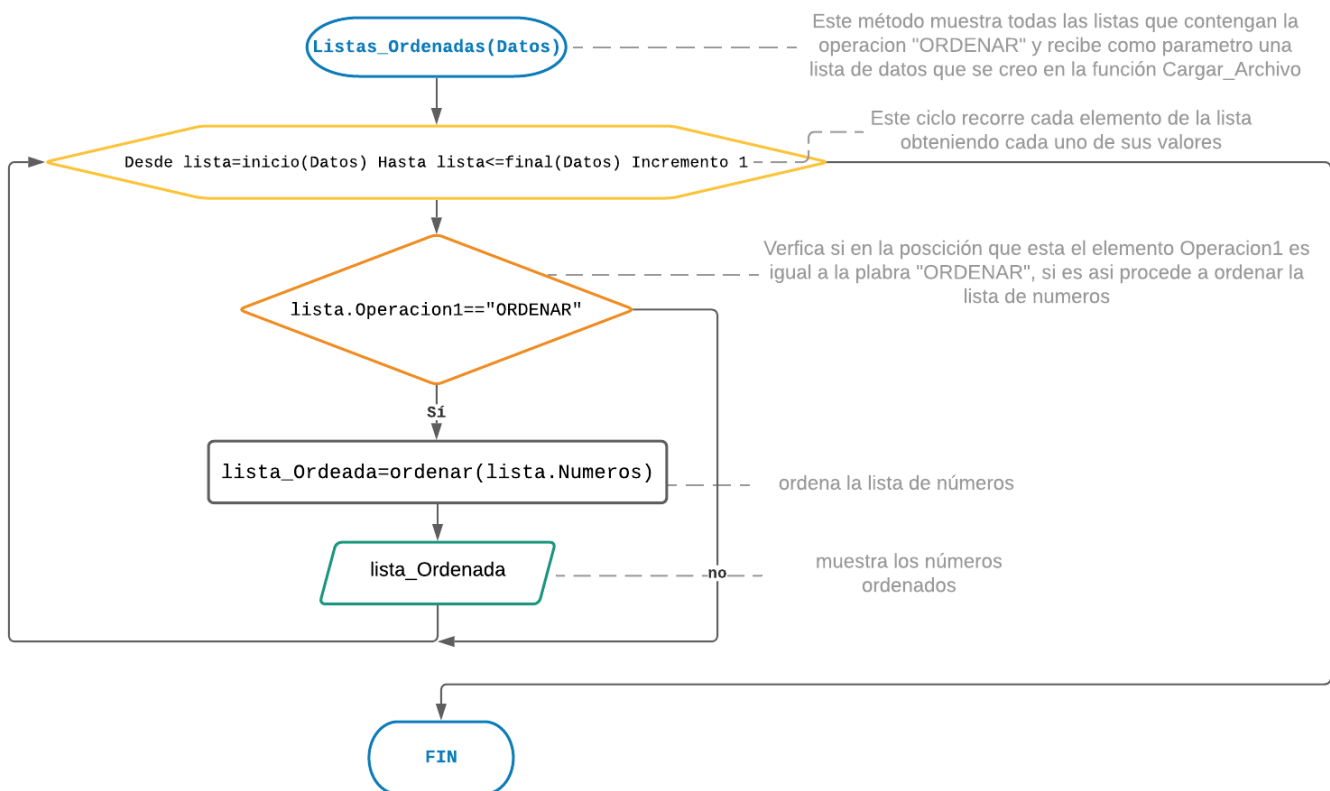
# Menú



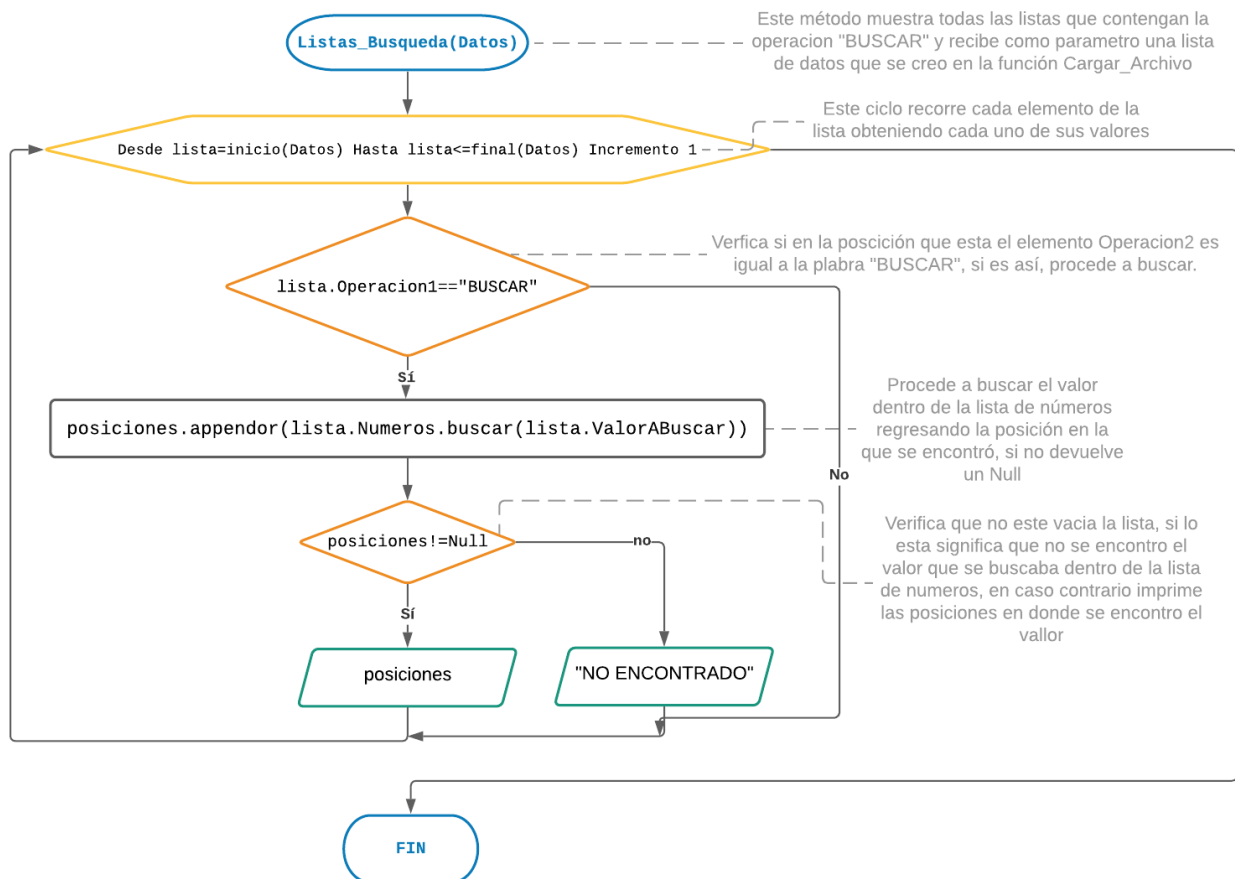
## Cargar archivo de entrada:



## Desplegar listas ordenadas:

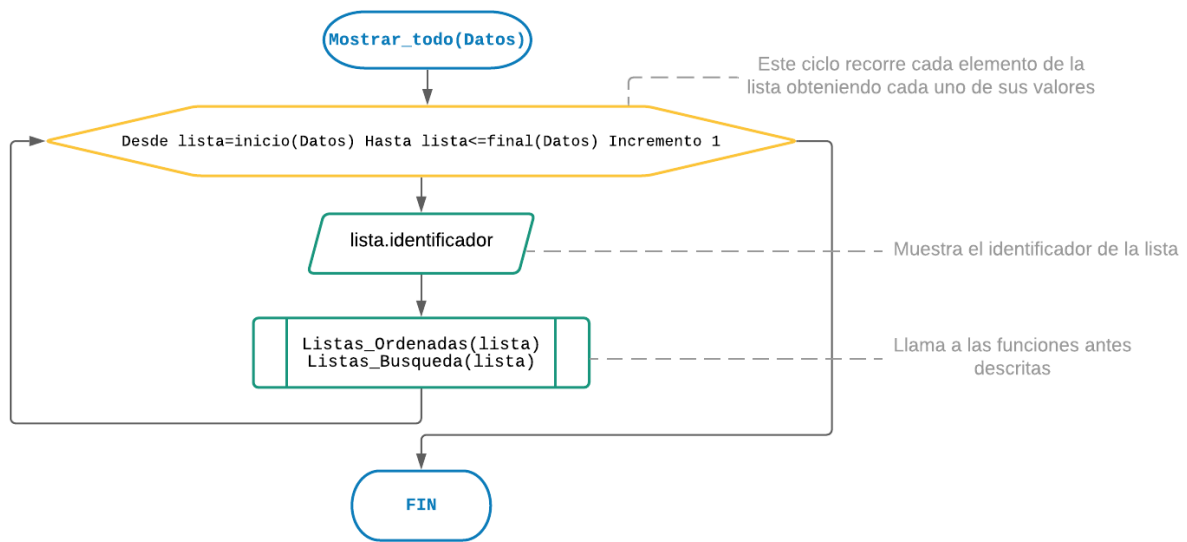


## Desplegar búsquedas:

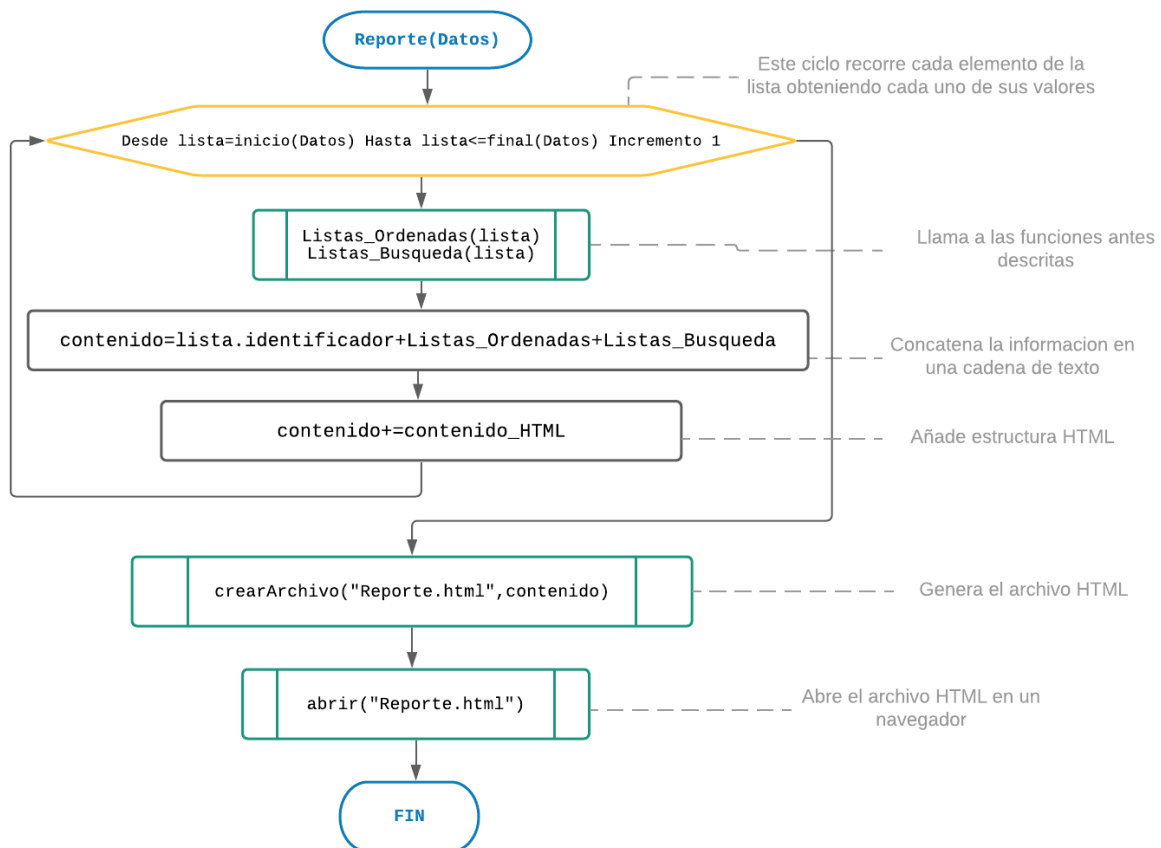




## Desplegar todas:



## Desplegar todas a archivo



# PARADIGMAS DE PROGRAMACIÓN UTILIZADAS

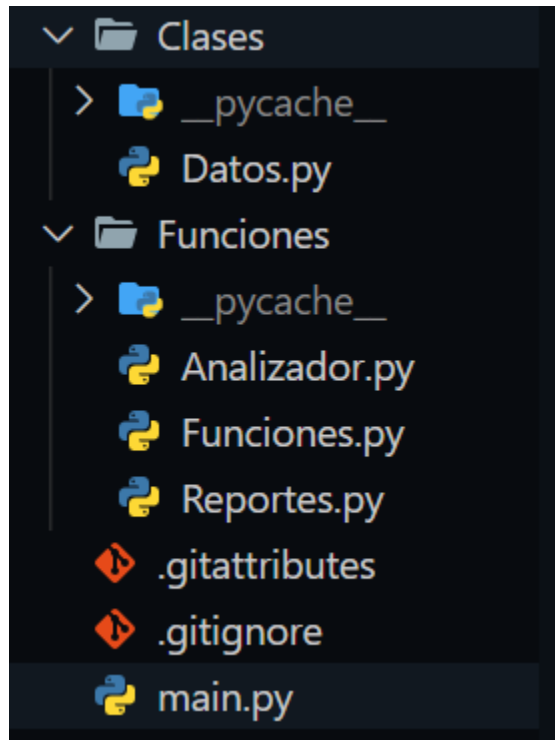
## Programación estructurada

```
1 def obtenerNumeros(txt):
2     #if txt.startswith(" "):
3         #txt=txt.replace(" ", "", 1)
4     global numeros
5     s=txt.replace(" ", "")
6     if 'B' in s:
7         s=s.replace("B", " B")
8     if '0' in s:
9         s=s.replace("0", " 0")
10    n=s.split(" ", 1)
11    numeros=n[0].split(",")
12    for x in range(len(numeros)):
13        numeros[x] = int(numeros[x])
14    #print(" > Lista_numeros = ", numeros)
15    obtenerOperaciones(n[1])
16
17    def menu():
18        opcion=0
19        while opcion!=6:
20            print("\n-----MENÚ PRINCIPAL-----\n")
21            print(" 1. Cargar archivo de entrada")
22            print(" 2. Desplegar listas ordenadas")
23            print(" 3. Desplegar búsquedas")
24            print(" 4. Desplegar todas")
25            print(" 5. Desplegar todas a archivo")
26            print(" 6. Salir\n")
27            opcion=int(input("- Ingrese una opción:\n > "))
28            switch={1:cargarArchivo, 2: listasOrdenadas, 3: busquedas, 4: todo, 5: desplegarTodo, 6: salir}
29            func=switch.get(opcion, "Opción inválida")
30            try:
31                func()
32            except:
33                print("\n > Opción inválida...")
34                input(" - PRESIONE ENTER PARA CONTINUAR...")
35 menu()
```

## Programación procedimental

```
1 def cargarArchivo():
2     obtenerArchivo()
3
4 def listasOrdenadas():
5     if ruta!="":
6         Funciones.listasOrdenadas(datos)
7     else:
8         print(" > ERROR: No se ha cargado ningún archivo")
9         input("\n- PRESIONE ENTER PARA CONTINUAR...")
10
11 def busquedas():
12     if ruta!="":
13         Funciones.busquedas(datos)
14     else:
15         print(" > ERROR: No se ha cargado ningún archivo")
16         input("\n- PRESIONE ENTER PARA CONTINUAR...")
17
18 def todo():
19     if ruta!="":
20         Funciones.delegarTodas(datos)
21     else:
22         print(" > ERROR: No se ha cargado ningún archivo")
23         input("\n- PRESIONE ENTER PARA CONTINUAR...")
24
25 def desplegarTodo():
26     if ruta!="":
27         Reportes.reporte(datos)
28     else:
29         print(" > ERROR: No se ha cargado ningún archivo")
30         input("\n- PRESIONE ENTER PARA CONTINUAR...")
31
32 def salir():
33     print(" > Saliendo...\n")
```

## Programación modular



```
from Funciones import Analizador
from Funciones import Funciones
from Funciones import Reportes
```

## Programación Orientada a Objetos

```
1 class Datos:
2     def __init__(self, linea, identificador, lista, operacion0, operacionB, nBuscar):
3         self.linea=linea
4         self.identificador=identificador
5         self.lista=lista
6         self.operacion0=operacion0
7         self.operacionB=operacionB
8         self.nBuscar=nBuscar
9
10    def getIdentifiador(self):
11        return self.identificador
12
13    def getListas(self):
14        return self.lista
15
16    def getOperacion0(self):
17        return self.operacion0
18
19    def getOperacionB(self):
20        return self.operacionB
21
22    def getNBuscar(self):
23        return self.nBuscar
```

## CONCLUSIÓN:

- Es necesario establecer un formato y estructura al archivo que se desea analizar y procesar ya que con ello se pueden crear algoritmos que ayudaran a realizar tales acciones.
- El explorador de archivos hace más fácil la búsqueda y carga de archivos.
- Los reportes permiten visualizar de forma detallada cierta información.