

Luis David Juarez Reyes  
201801106



# 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 ascendente .....	7
Desplegar listas ordenadas descendente .....	8
Desplegar lista .....	9
Generar Reporte .....	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 con extensión. lfp dado con un formato específico, el cual contiene un identificador de curso , lista de estudiantes y operaciones a realizar (parámetros), 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 y Os para mostrar un explorador de archivos y abrir automáticamente el reporte.

# 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:

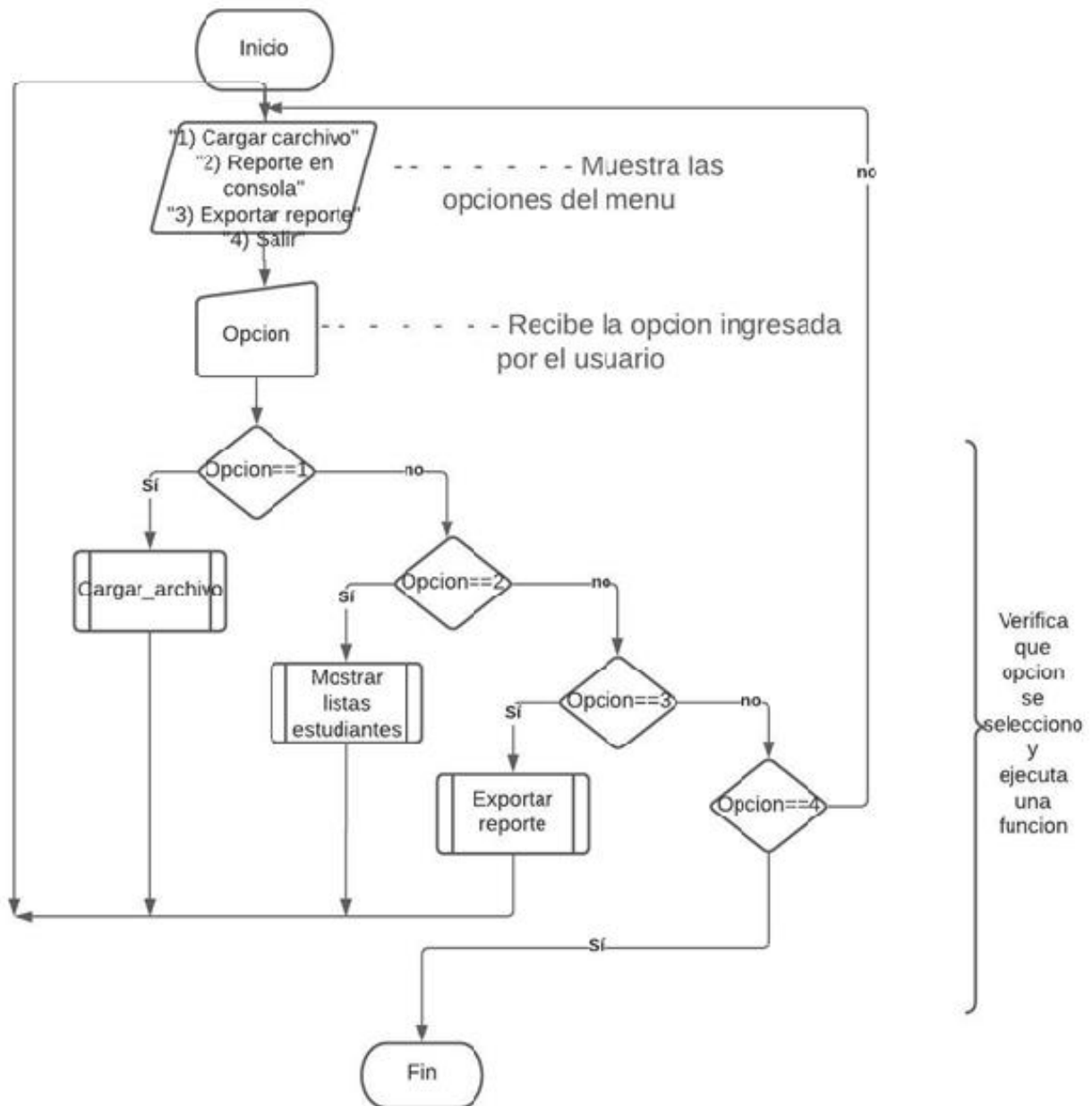
- Python 3.8.1
- Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- Privilegios de administrador para la instalación

## Linux:

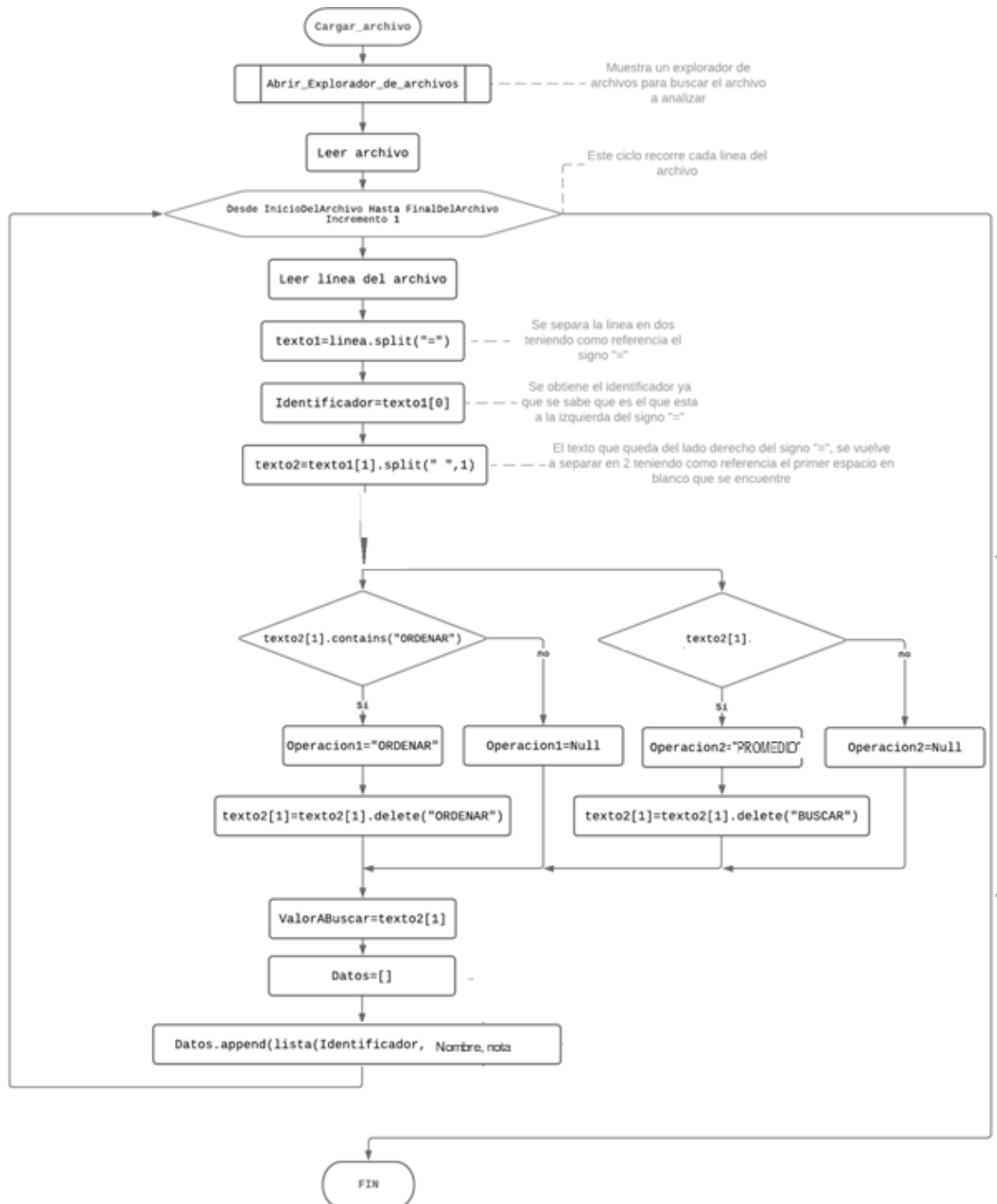
- Python 3.8.1
- Cualquier distribución de Linux.

En cualquiera de los sistemas que vaya a ejecutar la aplicación se debe contar con un navegador que soporte HTML 5, para poder visualizar el reporte.

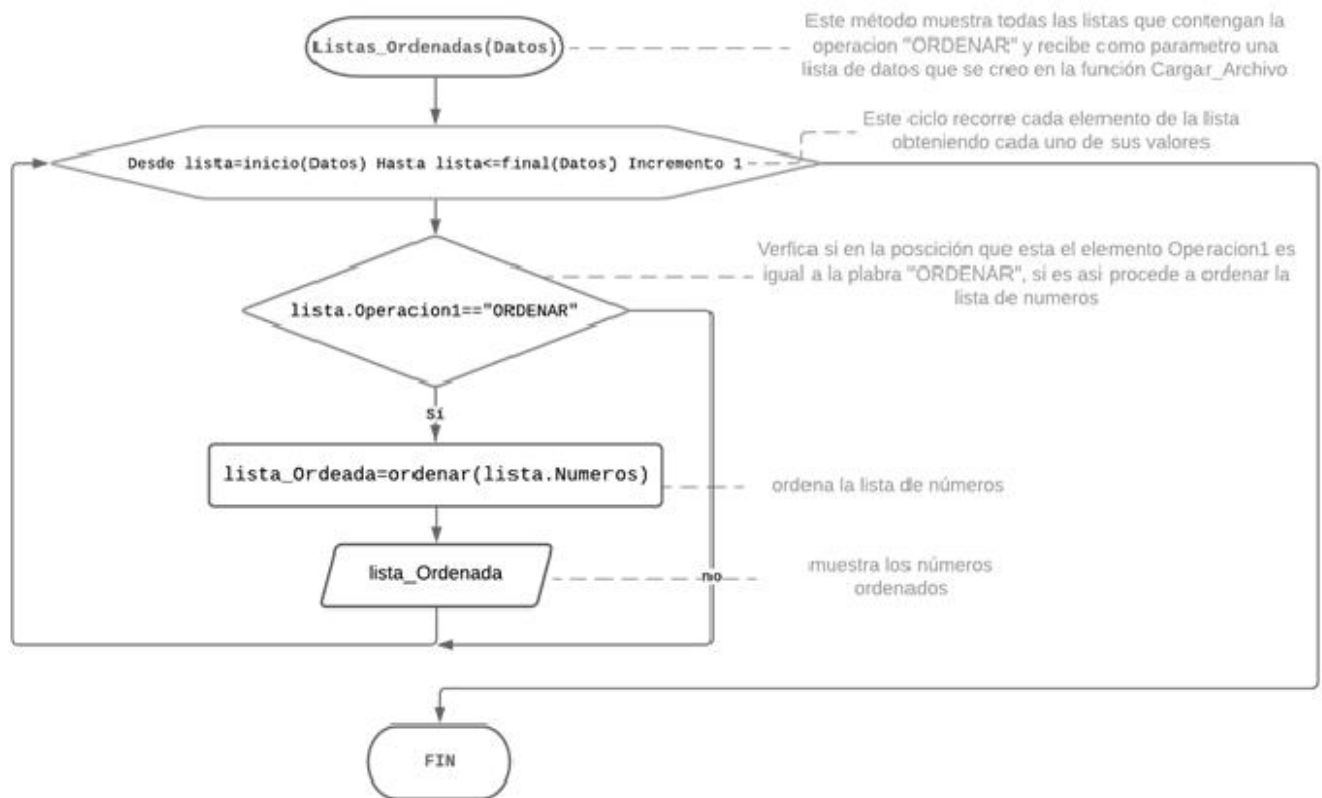
## Menú



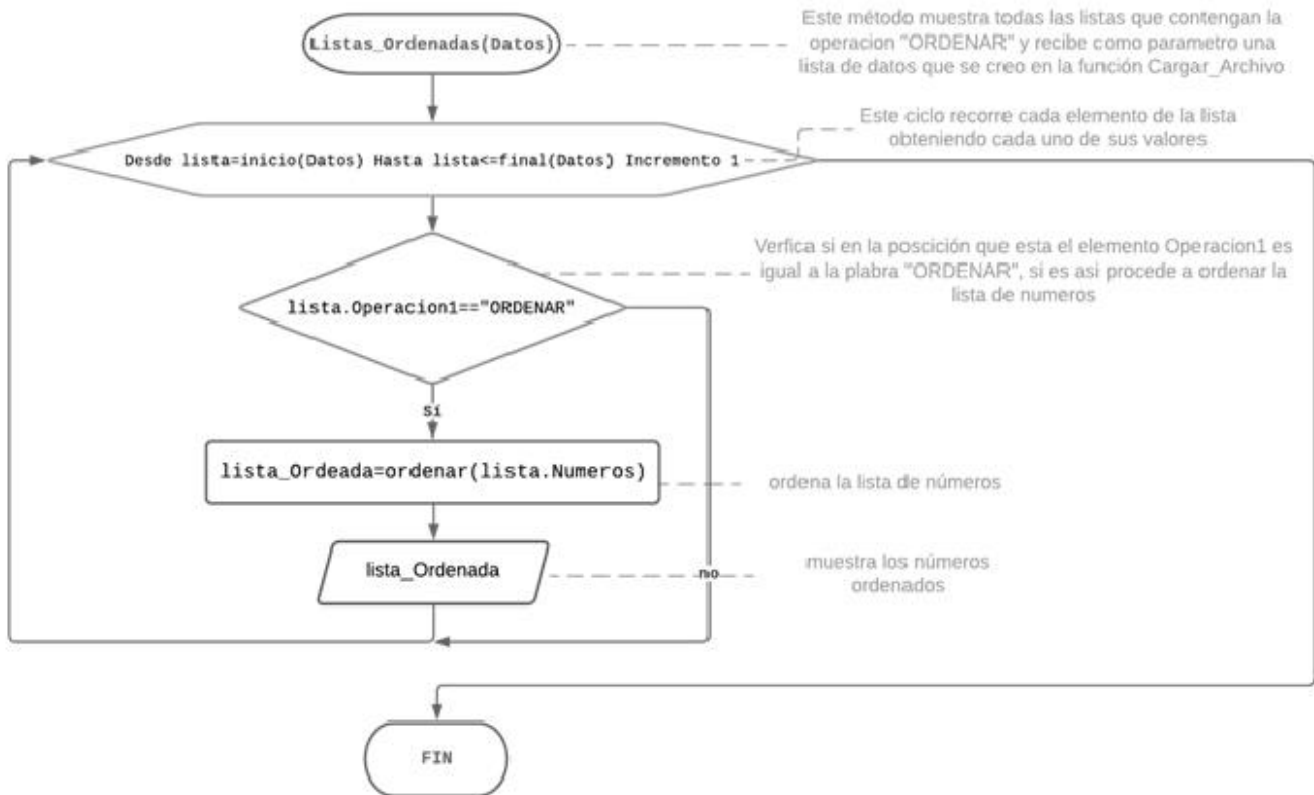
## Cargar archivo de entrada:



## Desplegar listas ordenadas ascendente:

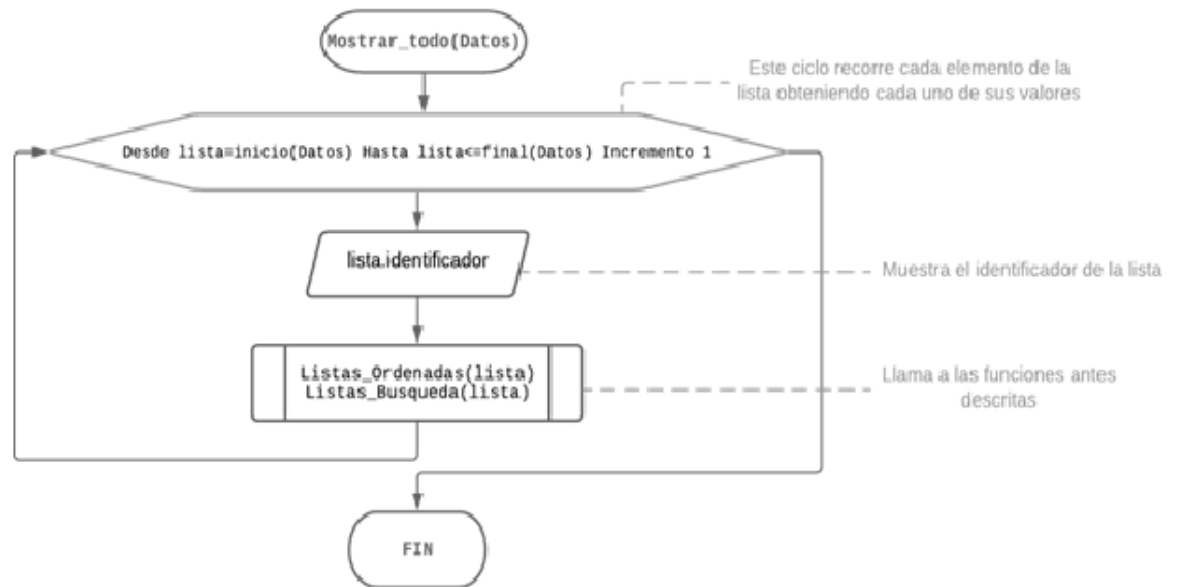


## Desplegar listas ordenadas descendente

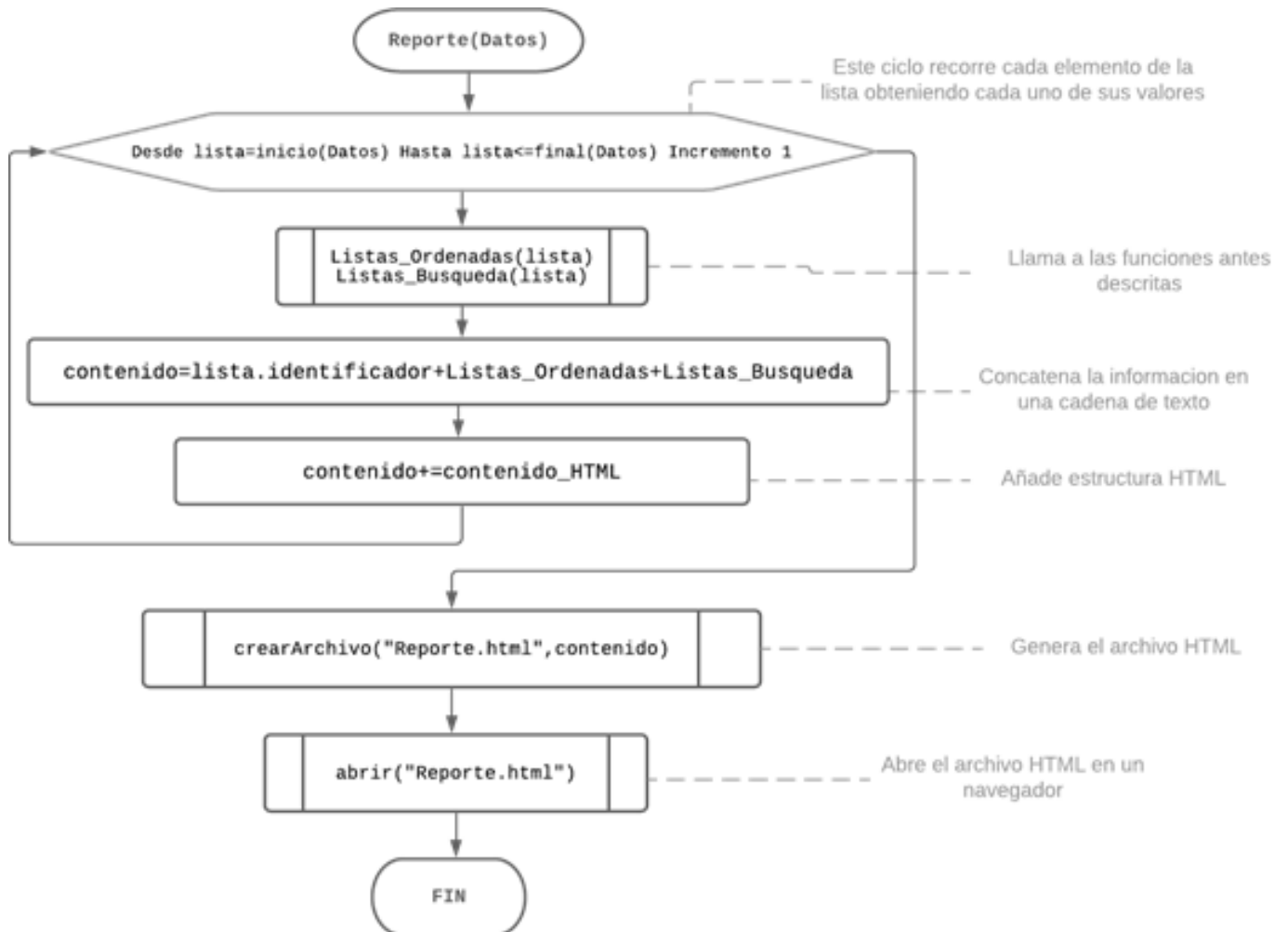




## Desplegar Lista:



## Generar Reporte



# PARADIGMAS DE PROGRAMACIÓN UTILIZADAS

## Programación estructurada

```
def Ordenar_Ac(boton):
    global notaMaxima
    no=[]
    nu=[]
    for i in alumnos:
        no.append(i.getNombre())
        nu.append(int(i.getNota()))

    for top in range(len(nu)-1,0,-1):
        for i in range(top):
            if nu[i]>nu[i+1]:
                aux = nu[i]
                aux2=no[i]

                nu[i] = nu[i+1]
                no[i] = no[i+1]

                nu[i+1] = aux
                no[i+1] = aux2

    notaMaxima[0]=str(no[(len(alumnos)-1)] )
    notaMaxima[1]=str(nu[(len(alumnos)-1)] )

    notaMinima[0]=str(no[0] )
    notaMinima[1]=str(nu[0] )

    if boton==1:
        global reporte
        reporte=reporte + PP1.titulotabla1 + "NOTA ORDENADA ASCENDENTEMENTE" +PP1.titulotabla2
        print("\n\t\t NOTA ORDENADA ASCENDENTEMENTE")
        tabla='' <table class="w3-table w3-striped w3-bordered w3-border">
        <thead class="w3-green">
        ...

        reporte=reporte+tabla
        tati='' <tr>
        <th scope="col"> NOMBRE</th>
        <th scope="col"> NOTA</th>
        </tr>'''

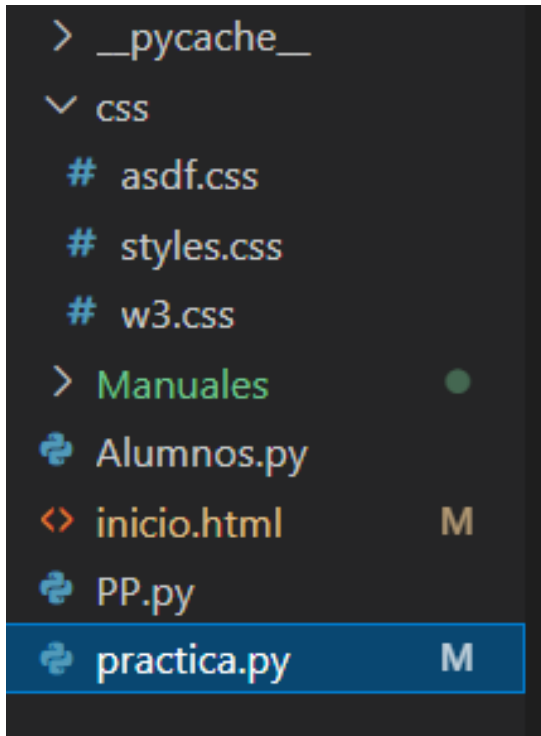
        ##tati es la variable para titutlo (NOMBRE , NOTA)
        reporte=reporte+tati

        for i in range(len(no)):
            print("Nombre : " + no[i]+ " Nota: " + str(nu[i]))
            sc1='<tr> <th scope="col"> '
            sc2='' </th>
            <th class="blue">'''

            sc22='' </th>
            <th class="red">'''
            sc3="</th> </tr>"
            if int(nu[i])>60:
                reporte=reporte+sc1+str(no[i])+sc2+str(nu[i])+sc3
            else:
                reporte=reporte+sc1+str(no[i])+sc22+str(nu[i])+sc3
```

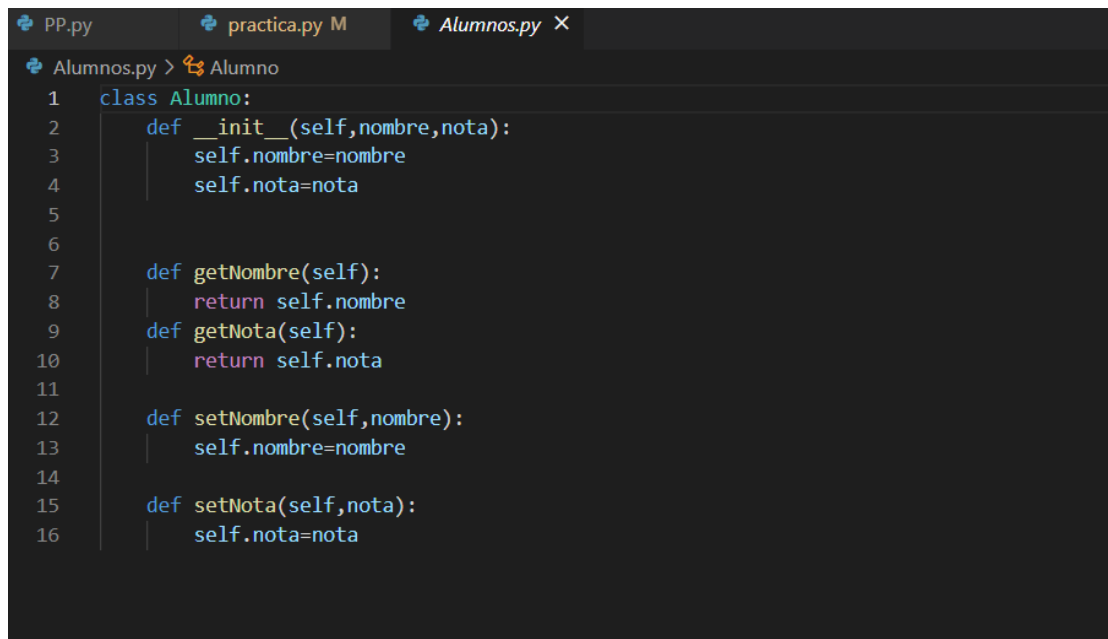
```
Imprimir()
for i in restricciones:
    if i=="ASC":
        Ordenar_Ac(1)
    elif i=="DESC":
        Ordenar_DC()
    elif i=="AVG":
        Promedio()
    elif i=="MIN":
        NMinima()
    elif i=="MAX":
        NMaxima()
    elif i=="APR":
        Aprobados()
    elif i=="REP":
        Reprobados()
```

## Programación modular



```
from Alumnos import Alumno
from PP import PP1
```

## Programación Orientada a Objetos



## 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.