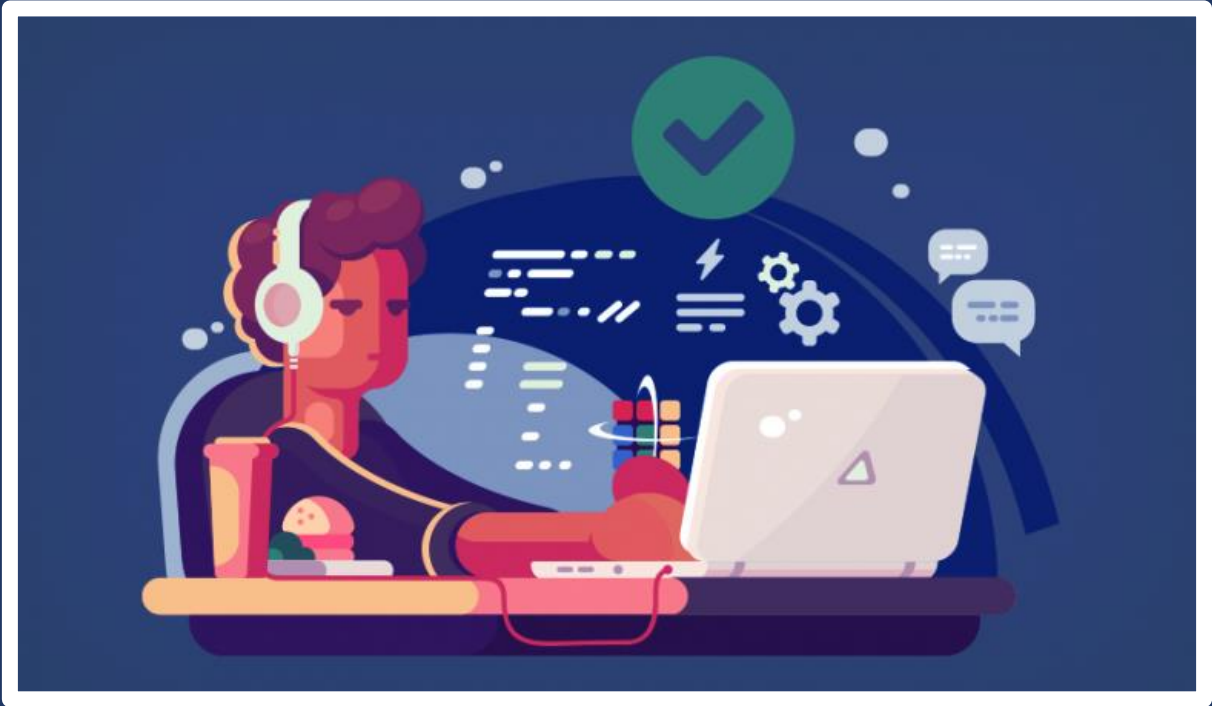
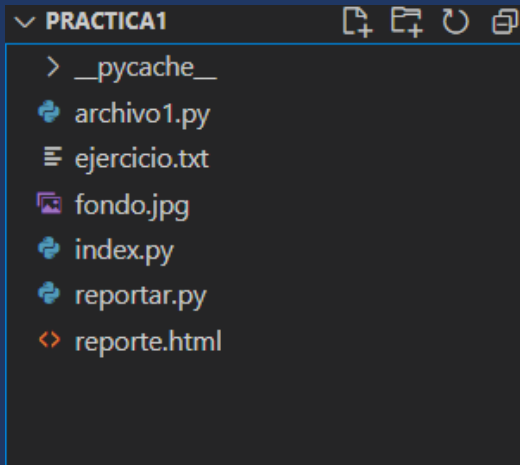


# MANUAL TÉCNICO



# Archivos creados:



**-index.py**

contiene el menú:

```
def mostrar_datos():
    print("NOMBRE DEL CURSO: " ,sep1)
    print( "TOTAL DE ALUMNOS: " ,contar_elementos(contenido_nombrepunteo) )

    if "ASC" in lista3:
        print("ORDEN ASCENDENTE")
        ordenarAs(notas)
        print(notas)
    if "DESC" in lista3:
        print("ORDEN DESCENDENTE")
        ordenarDes(notas)
        print(notas)

    if "AVG" in lista3:
        promedio(notas)

    if "MIN" in lista3:
        minimo(notas)

    if "MAX" in lista3:
        maximo(notas)

    if "REP" in lista3:
        reprobado(notas)

    if "APR" in lista3:
        aprobar(notas)
```

# El analizador del archivo:

```
def cargarArchivo():

    global contenido_archivo
    global contenido_nombrepunteo, notas
    global sep1, sep2
    global lista2, lista3, listanueva

    contenido_archivo.clear()
    extension = [".py", ".lfp"]

    archivo = eg.fileopenbox(msg="Abrir archivo", title="Control: Practica Lenguajes", default="", filetype=extension)
    efe = open(archivo, "r", 5, "utf-8")
    for linea in efe.readlines():
        if linea != "\n":
            contenido_archivo.append(linea)
            print(Fore.RED + linea.rstrip())

    efe.close()

    fila=0
    col=0

    contenido_archivo[-1]=contenido_archivo[-1] + "#"

    sep1, fila, col = splitear(contenido_archivo, "=", fila, col)
    lista1 = alumnos(sep1, "=")
    #print(lista1)
    print(sep1)

    while True:
        sep2, fila, col = splitear(contenido_archivo, "<", fila, col)

        sep2, fila, col = splitear(contenido_archivo, ">", fila, col+1)

        lista2 = alumnos(sep2, ";")
        #print(lista2[0])
        listanueva = Estudiante(lista2[0], lista2[1])

        print(sep2)

        #print(lista2)
        contenido_nombrepunteo.append(lista2) #Para contar cuantos alumnos hay
```

# Archivos importados: Variables globales

```
from archivol import Estudiante, alumnos, aprobar, ordenarAs, ordenarDes, reprobar, splitear, contar_elementos, promedio, maximo, minimo
from colorama import init
from colorama import Fore, Back, Style
import easygui as eg

contenido_archivo=[]
contenido_nombrepunteo = []
notas= []

init(autoreset=True)
```

# -Archivo1.py

Contiene todos los métodos a utilizar en el index.py

## Splitear()

```
def splitear(cadena,simbolo,fila,columna):
    #print("Si llego a mínimo")
    temporal = ""
    listaTemporal = []

    for i in range(fila,len(cadena)): #Obteniendo el tamaño del contenido del arreglo
        linea = cadena[i] # capturo la línea de la lista de contenido por el índice i
        if fila != i:
            columna=0 #Para resetear la línea
            fila=i
        if linea != "":
            for c in range(columna,len(linea)) : #Obteniendo el tamaño del cadena
                caracter=linea[c] #capturo el caracter por el índice c
                if caracter==simbolo :
                    #listaTemporal.append(temporal.strip())
                    #temporal = ""
                    return temporal,i,c
                else:
                    if caracter != "\n" :
                        temporal +=caracter

            #if temporal.strip() != "":
            #    listaTemporal.append(temporal.strip())

    # return listaTemporal # Necesito devolver dos cosas , variable token y un índice
    #-----
```

# alumnos()

```
def alumnos (cadena,simbolo):  
    #listaNueva  
  
    #Laura, 17  
    #datos = ['Laura' , '17']  
    lista = []  
    temporal=""  
    for s in cadena:  
        if s==simbolo:  
  
            lista.append(temporal)  
            temporal=""  
        else:  
            temporal +=s  
    lista.append(temporal)  
  
    return lista
```

# ordenarAs ()

```
def ordenarAs(unalista):  
  
    for numPasada in range(len(unalista)-1,0,-1):  
        for i in range(numPasada):  
            if unalista[i]>unalista[i+1]:  
                temp = unalista[i]  
                unalista[i] = unalista[i+1]  
                unalista[i+1] = temp
```

# OrdenarDes()

```

def ordenarDes(unalista):
    for numPasada in range(len(unalista)-1,0,-1):
        for i in range(numPasada):
            if unalista[i]<unalista[i+1]:
                temp = unalista[i]
                unalista[i] = unalista[i+1]
                unalista[i+1] = temp

```

# Contar\_elementos()

```

def contar_elementos(lista):
    count=0
    for elementos in (lista):
        count+=1
    return count
#-----

```

# Aprobar() y Reprobar()

```

def aprobar(lista):
    notas_aprobadas=[]
    lista_elementos= [int(x)for x in lista]
    for x in lista_elementos:
        if x >= 61 :
            notas_aprobadas.append(x)
    print("ESTUDIANTES APROBADOS: ",len(notas_aprobadas))

#-----
def reprobar(lista):
    notas_reprobadas =[]
    lista_elementos= [int(x)for x in lista]
    for x in lista_elementos:
        if x < 61:
            notas_reprobadas.append(x)
    print("ESTUDIANTES REPROBADOS: ",len(notas_reprobadas))
#-----

```

# Máximo () y Mínimo()

```
def maximo(lista):  
    lista_elementos= [int(x)for x in lista]  
    print("NOTA MÁXIMA: ",max(lista_elementos))  
  
#-----  
def minimo(lista):  
    lista_elementos= [int(x)for x in lista]  
    print("NOTA MÍNIMA: ",min(lista_elementos))  
#-----
```

# promedio()

```
def promedio(lista):  
    listasuma= [int(x)for x in lista]  
    sumarelementos = sum(listasuma)  
    num_elementos=len(lista)  
    media= sumarelementos/num_elementos  
    #suma=0  
  
    #for i in listasuma:  
    |    #suma+=listasuma[i]  
    print("PROMEDIO:", round(media,2))
```

# -reportar.py

## Contiene el archivo html concatenado

```
def reportar(lista):
    report =webbrowser.open("reporte.html","w","5","utf-8")
    report.write(
        "<!DOCTYPE html>"
        + "<html lang='en'">"
        + "<head>"
            + "<meta charset='UTF-8'">"
            + "<meta http-equiv='X-UA-Compatible' content='IE=edge'">"
            + "<meta name='viewport' content='width=device-width, initial-scale=1.0'">"
            + "<title>REPORTE ALUMNOS</title>"
        + "</head>"
        + "<style>"
            + "body {"
            + "background-image: url('fondo.jpg');"
            + "}"
            + "table, th {"
            + "border: 1px solid black;"
            + "background-color: azure;"
            + "font-size: xx-large;"
            + "font-style: oblique;"
        + "}"
        + "h1{"
            + "background-color: rgb(220, 207, 20);"
            + "font-style: italic;"
        + "}"
        + "</style>"
        + "<body >"

            + "<h1 style='text-align:center'">REPORTE ESTUDIANTES </h1>"
            + "<table Align='center'">"
                + "<tr> "
                    + "<th> Nombre </th>"
                    + "<th>Punteo </th>"
                + "</tr>"

    )
```



# -reporte.html

## Contiene la estructura de la página web

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>REPORTE ALUMNOS</title>
</head>
<style>
  body {
    background-image: url('fondo.jpg');
  }
  table, th {
    border: 1px solid black;
    background-color: azure;
    font-size: xx-large;
    font-style: oblique;
  }
  h1 {
    background-color: rgb(220, 207, 20);
    font-style: italic;
  }
</style>
<body >

  <h1 style="text-align:center">REPORTE ESTUDIANTES </h1>
  <table Align="center">
    <tr>
      <th> Nombre </th>
      <th>Punteo </th>
    </tr>
  </table>
</body>
</html>
```