

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

LENGUAJES FORMALES Y DE PROGRAMACION

SECCION "B"

PRACTICA #1



---

MANUAL DE TECNICO

---

JUAN PABLO GONZALEZ LEAL

201901374

GUATEMALA FEBRERO 2021

## SOFTWARE LECTURA DE ARCHIVO

Es un muy impórtate este tipo de software para la organización de datos, no solo en el ámbito empresarial sino para un desarrollador de aplicaciones o software, la organización de datos.

|  |   |
|--|---|
| <p><b>main.py</b></p> <pre>def imprimir():     while True:         print('_____ Menu _____')         print('1. Cargar archivos')         print('2. Desplegar Listas ordenadas')         print('3. Desplegar busquedas')         print('4. Desplegar todas')         print('5. Desplegar todas a archivo')         print('6. Salir')         print('_____')         opcion=int(input('ELIJA UNA OPCION'))         if opcion==1:             lista.clear()             listas_ordenadas.clear()             lista_busquedas .clear()             cargar_archivo()         elif opcion==2:             desplegar_ordenados()         elif opcion==3:             desplegar_busqueda()         elif opcion==4:             desplegar_todas()         elif opcion==5:             desplegar_todas_archivo()         elif opcion==6:             break</pre> | <p>Contamos con una función para la selección de una opción, cada opción se define por un número, este establecido por el usuario, limpiamos las listas a las cual se le agregaran los datos esto para que no existan residuos de datos pasados. dependiendo la opción nos dirige a una función que realiza la acción que se desea.</p>   |
| <pre>def cargar_archivo():     contador = 0     ruta=filedialog.askopenfilename(title="abrir")     archivo=open(ruta)     for linea in archivo.readlines():         archivo_sin_espacios=linea.replace(" ","")         analizador.analizar(archivo_sin_espacios,contador)         contador += 1     archivo.close()     print('Datos Cargados')  def desplegar_ordenados():     for i in listas_ordenadas:         print(i)  def desplegar_busqueda():     for j in lista_busquedas:         print(j)  def desplegar_todas():     for k in lista:         print(k)  def desplegar_todas_archivo():     nombre_archivo=input('Nombre archivo:')     crear_html.crear(nombre_archivo)  imprimir()</pre>  | <p>Creamos el procedimiento de las funciones. En la función cargar_archivo, se hace el llamado una ventana emergente para seleccionar uno, este va direccionar cada línea a un archivo .py a una función analizar.</p> <p>Las demás funciones solo imprimen resultados del archivo analizador.py</p> <p>La función desplegar_todas_archivo se pasa como parámetro el nombre del archivo .HTML a crear</p> |
| <p><b>Nombre de métodos Utilizados:</b></p> <p><b>Imprimir():</b></p> <p><b>Cargar_archivo():</b></p> <p><b>Desplegar_ordenados():</b></p> <p><b>Desplegar_busqueda():</b></p> <p><b>Desplegar_todas():</b></p> <p><b>Desplegar_todas_archivo():</b></p>   |   |
| <p><b>analizador.py</b></p>  |   |

|   |  |
|---|--|
| <pre> listaABC=['A','B','C','D','E','F','G','H','I','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'] lista=[] listas_ordenadas=[] lista_busquedas=[]  def analizar(cadena,contador):     #array_identificador=cadena.split('=')     lista_numeros=[]     contador_posicion=0     posicion=""     cajon=""     cajon1=""     cajon2=""     cajon3=""     cajon4=""     cajon5=""     cajon6=""     paso1=True     paso2=False     paso3=False     paso4=False     for i in cadena:         caracter=i         if paso1==True:             if caracter=="=":                 paso1=False                 paso2=True                 cajon=cajon+":"             else:                 cajon=cajon+caracter         elif paso2==True:             verificar=caracter in listaABC             if verificar==True:                 paso2=False                 if caracter=="0":                     cajon3=cajon3                     paso3=True                 elif caracter=="B":                     cajon4=cajon4                     paso4=True             else:                 if caracter==" ":                     lista_numeros.append(int(cajon2))                     cajon2=""                 else:                     cajon2=cajon2+caracter         elif paso3==True:             if caracter=="B":                 paso3=False                 paso4=True             else:                 cajon3="ORDENADOS"         elif paso4==True:             verificar2=caracter.isdigit()             if verificar2==True:                 cajon6=cajon6+caracter             else:                 cajon5="BUSQUEDA POSICIONES="      lista_numeros.append(int(cajon2))     if cajon5=="BUSQUEDA POSICIONES=":         for x in lista_numeros:             intStr=int(cajon6)             contador_posicion += 1             if x==intStr:                 posicion=posicion+str(contador_posicion-1)+","             if posicion=="":                 cajon2=str(lista_numeros)                 posicion="NO ENCONTRADO"             cajon6=posicion      if cajon3=="ORDENADOS":         ordenar=sorted(lista_numeros)         cajon2=str(ordenar)         lista_numeros.clear()      lista.append(str(contador)+" . "+cajon+cajon3+"="+cajon2+cajon4+cajon5+cajon6)      if cajon3=="ORDENADOS":         listas_ordenadas.append(str(contador)+" . "+cajon+cajon3+"="+cajon2)      if cajon5=="BUSQUEDA POSICIONES=" and posicion!="NO ENCONTRADO":         lista_busquedas.append(str(contador)+" . "+cajon+cajon5+cajon6) </pre> | <p>Se crea una lista con el alfabeto</p> <p>Y se hacen 3 listas mas para guardar los datos de cada opción del menú,</p> <p>Se realiza un método a la cual se le pasa la línea que esta leyendo el archivo con la posición de la línea , creamos booleanos en falsos y a uno le asignamos true para ir agregando datos y hacer el remplazo cuando se llegue a la condición dando valor falso al paso anterior y true al paso siguiente</p> <p>Realizando cada proceso se puede observar que en algunos elif se agregan datos a una lista_nummeros esto para pasar de castear los datos y utilizar un sorted para el ordenamiento de los datos.</p> <p>Hacemos 2 condiciones si existe la palabra búsqueda de posiciones se pasara a realizar la búsqueda esta condición antes de la condición de ordenar los datos para seguir las reglas del enunciado.</p> <p>Al finalizar todo este proceso se agregan a las listas que se mostraran al ser llamas en el menú.</p> |
| <p><b>Nombre de métodos Utilizados:</b></p> <p><b>Analizar(cadena,contador):</b></p>  |  |
| <p><b>crear_html.py</b></p>   |  |

```

from analizador import *
def crear(nombre):
    crear=open(nombre+".html","w")
    crear.write("<html lang='en'>")
    crear.write("<head>")
    crear.write("<meta charset='UTF-8'>")
    crear.write("<meta name='viewport' content='width=device-width, initial-scale=1.0'>")
    crear.write("<link rel='stylesheet' href='https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css'>")
    crear.write("<title>"+nombre+"</title>")
    crear.write("</head>")
    crear.write("<body>")
    crear.write("<div class='container'><h1>Tabla de datos</h1></div>")
    crear.write("<div class='container px-4'><div class='row gx-5'><div class='col-sm-6'>")
    crear.write("<div class='container'><div class='row'><div class='col-sm-6'>")

    for a in lista:
        crear.write("<tr><td>"+a+" </td></tr>")

    crear.write("</tbody></table><br>")
    crear.write("<table class='table table-striped'><thead><tr><th scope='col'>")
    for b in listas_ordenadas:
        crear.write("<tr><td>"+b+" </td></tr>")

    crear.write("</tbody></table><br>")
    crear.write("<table class='table table-striped'><thead><tr><th scope='col'>")
    for c in lista_busquedas:
        crear.write("<tr><td>"+c+" </td></tr>")

    crear.write("</tbody></table></div></div></div>")
    crear.write("<script src='https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js'>")
    crear.write("<script src='https://code.jquery.com/jquery-3.4.1.slim.min.js'>")
    crear.write("<script src='https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js'>")
    crear.write("</body>")
    crear.write("</html>")
    crear.close()
    print('archivo creado')

```

Se recibe por parámetro el nombre que se le dio al archivo, creando y escribiendo la estructura de HTML , utilizamos bootstrap para realizar las tablas y un for para cada lista imprimirla y crear una fila.

**Nombre de métodos Utilizados:**

**Crear(nombre):**

**Lógico:** En la realización de métodos y condiciones al utilizar if .

**Imperativo:** Al inicializar listas vacías y variables tanto de tipo Sting como boléanos.

**Dirigido por eventos:** Al hacer una entrada de datos por el usuario tanto al escribir un numero a la opción del menú, como a la hora de escribir el nombre del archivo que se creara como HTML.

**Orientado a aspectos:** El programa o software se divide en módulos o archivos de distinto nombre realizando cada archivo actividades diferentes.