Manual técnico

Dentro del programa se utilizo un el paradigma orientado a objetos para almacenar los datos, dentro del programa se utilizo la clase “mochilero” para almacenar y ordenar los datos sin importar los comandos que se encontraran en la cadena de caracteres.

```
listaMochileros =[]

class mochilero():
    def __init__(self):
        self.numbersNoOrdered = []
        self.numbersOrdered = None
        self.title = ""
        self.positions = None
        self.number_wanted=None
```

Para reconocer la cadena de caracteres se creó un autómata con una gramática de tipo 3, donde:

T= ["ė",",","=","\n","\t"] son los símbolos terminales ė=espacio en blanco

A=[a-zA-Z] son las letras aceptadas

N= [0-9] son los dígitos aceptados

Dentro del autómata se colocaron palabras reservadas que fueron los comandos “ordenar y buscar”.

Dentro de la lógica del programa el autómata va recopilando carácter por carácter (excepto los espacios en blanco) y al reconocer el primer carácter si esta en el conjunto A lo categoriza como nombre, si esta en el conjunto N lo categoriza como número. Si encuentra un símbolo terminal identifica que tipo de categoría esta y lo guarda dentro de mochilero según la sintaxis reconocida.

```
elif type == "nombre":

    if letter.lower() in A or letter in N:
        group = group + letter
    elif letter in T or letter == line[len(line)-1]:

        #COMANDO
        if group.lower() in C:
            type = "comando"

            if group.lower() == "ordenar":
                #numbersOrdered=numbers.copy()
                #numbersOrdered.sort()
                numbersOrdered = Burbuja(numbers.copy())

                mochila.numbersOrdered=numbersOrdered

                group = ""
                type = ""


        elif letter == "=":
            title = group
            mochila.title = title

            group = ""
            type = ""
```

De igual manera recolecta todos los números. Cuando el autómata detecta que los caracteres que recolecto coinciden con las palabras reservadas o los comandos efectúa la operación de acuerdo con el comando

```
elif type == "comando":
    if group.lower() == "buscar":...
```

```
if letter == "\n" or indice+1==len(line):

    if group.lower() in C:
        type = "comando"

        if group.lower() == "ordenar":...


    listaMochileros.append(mochila)
    group = ""
    type = ""
    numbers=[]
```