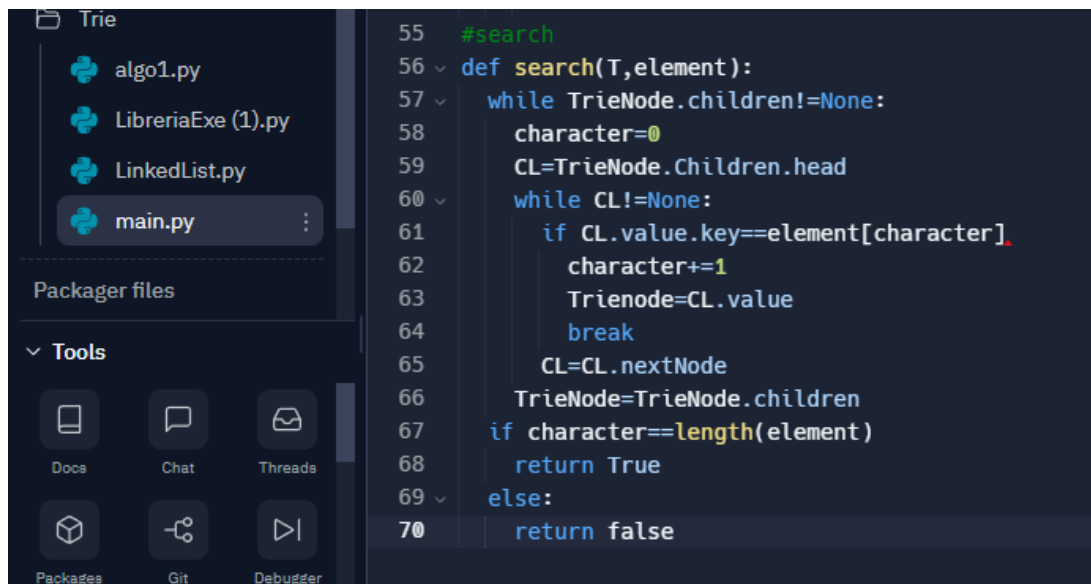# Algoritmos II

Trabajo Practico arboles Nºarios-Trie

1) a) Insert:

```python
15   def SearchCharacter(level,element):
16     for i in range(0,length(element)-1):
17       if i==(level-1):
18         c=element[i]
19         break
20     return c
21   #insert
22   def insert(T,element):
23     level=1
24     if TrieNode.children==None:
25       TrieNode.children=LinkedList()
26       CL=Linkedlist()
27       CL=None
28       T=Trienode()
29       T.key=SearchCharacter(level,element)
30       add(TrieNode.children,T)
31       CL=TrieNode.children.head
32       level+=1
33     else:
34       CL=TrieNode.children.head
35       while CL!=None:
36         if CL.value.key==c:
```

```python
36         if CL.value.key==c:
37           TrieNode=CL.value
38           CL.head=TrieNode
39           break
40         CL=CL.nextNode
41       if CL==None:
42         T=TrieNode()
43         T.key=SearchCharacter(level,element)
44         add(TrieNode.children,T)
45         CL=TrieNode.children.head
46       level+=1
47     for i in range (1,length(element)-1):
48       T=TrieNode()
49       T.key=SearchCharacter(level,element)
50       insert(CL,T,i)
51       T.parent=searchCurrent(CL, element[i-1])
52       if i==(length(element)-1):
53         T.isEndOfWord=True
```

B) Search:

```
55   #search
56   def search(T,element):
57     while TrieNode.children!=None:
58       character=0
59       CL=TrieNode.Children.head
60       while CL!=None:
61         if CL.value.key==element[character]
62           character+=1
63           Trienode=CL.value
64           break
65         CL=CL.nextNode
66       TrieNode=TrieNode.children
67     if character==length(element)
68       return True
69     else:
70       return false
```