Diccionarios con defaultdict

Algoritmo de construcción de diccionarios con defaultdict donde los valores son acumuladores

Ejemplo: Para la lista lista=['a','b','c','a','b','b','z','a','b','c'], contar cuantas veces aparece cada valor'

```
dic=dict()
for letra in  lista:
    if letra not in dic:
        dic[letra]=0          //Elemento neutro de la suma
    dic[letra]+=1
```

Con defaultdict:

```
from collections import defaultdict
dic=defaultdict(int)
for letra in  lista:
    dic[letra]+=1
```

Resultado: dic →{ 'a': 3, 'b': 4, 'c': 2, 'z': 1}

Cuando se accede a una clave que no existe no da error y en este caso devuelve cero: dic['h']→0

Diccionarios con defaultdict

Algoritmo de construcción de diccionarios con defaultdict donde los valores son contenedores

Ejemplo: Si tenemos el tipo Par=namedtuple("Par", "letra, número") y lista=[Par("z",21),Par("a",62),Par("J",7),Par("b",56),Par("c",90),Par("z",21),Par("a",1),Par("b",10),Par("b",2),Par("a",21)], obtener un diccionario que a cada letra le haga corresponder una lista con los valores asociados

```
dic=dict()
for p in lista:
if p.letra not in dic:
    dic[p.letra]=list()
dic[p.letra].append(p.número)
```

Con defaultdict:

```
from collections import defaultdict
dic=defaultdict(list)
for p in lista:
    dic[p.letra].append(p.número)
```

Resultado: dic → {'z': [21, 21], 'a': [62, 1, 21], 'J': [7], 'b': [56, 10, 2], 'c': [90]}

Cuando se accede a una clave que no existe no da error y en este caso devuelve [ ]: dic['A'] → [ ]