

NIVEL 2

DICCIONARIOS: EJERCICIO



EJERCICIO



Pensemos: cómo contar la cantidad de ocurrencias de cada dígito (del 0 al 9) en un número de 10 cifras usando diccionarios.

✓ Si el número es 8384724289, tenemos que:

El 8 aparece 3 veces

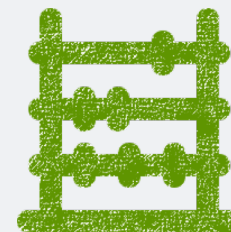
El 3 aparece 1 vez

El 4 aparece 2 veces

El 7 aparece 1 vez

El 2 aparece 2 veces

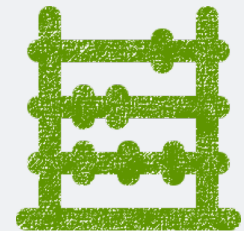
El 9 aparece 1 vez



Puedes verificar tus resultados usando la terminal presente en la actividad “Manos a la obra: Ocurrencias de cada dígito en un número usando diccionarios” en Brightspace

PISTAS PARA LA SOLUCIÓN

- ✓ Use un diccionario, en el cual las claves sean los dígitos del 0 al 9 y los valores sean las ocurrencias de cada dígito en el número de 10 cifras
- ✓ Extraiga uno a uno los dígitos del número y vaya acumulando las ocurrencias en el diccionario



VEAMOS CUATRO VERSIONES



- **Versión 1:** sin funciones (todo en el programa principal)
- **Versión 2:** con una función que retorna un diccionario modificado
- **Versión 3:** con una función que retorna un entero y modifica el diccionario por dentro de la función
- **Versión 4:** con una función que retorna un entero y modifica el diccionario por dentro de la función pero sin condicional

VERSIÓN 1: SIN FUNCIÓN

Se repite 10 veces este bloque de instrucciones:

1. Se saca el último dígito del número
2. Se actualiza el diccionario con las ocurrencias del dígito: si el dígito ya se encontraba en el diccionario, se incrementa en 1 sus ocurrencias. Si no, se ingresa con 1 ocurrencia (la clave es el dígito y el valor es 1)
3. Se recorta el número

```
EjemploConteoDigitos_version1.py
1 #Se pide el número al usuario
2 numero = int(input("Digite el número de 10 cifras: "))
3
4 #Se crea el diccionario
5 conteo = {}
6
7 #1: Se saca el último dígito del número
8 digito = numero%10
9 #Se actualiza el diccionario con las ocurrencias del dígito
10 if (digito in conteo):
11     conteo[digito] += 1
12 else:
13     conteo[digito] = 1
14 #Se recorta el número
15 numero = numero//10
16
17 #2: Se saca el último dígito del número
18 digito = numero%10
19 #Se actualiza el diccionario con las ocurrencias del dígito
20 if (digito in conteo):
21     conteo[digito] += 1
22 else:
23     conteo[digito] = 1
24 #Se recorta el número
25 numero = numero//10
26
27 # Y LO MISMO 8 VECES MÁS ...
```

Resultado de la ejecución

```
Terminal de IPython
Terminal 3/A
Digite el número de 10 cifras: 8384724289
{9: 1, 8: 3, 2: 2, 4: 2, 7: 1, 3: 1}
```

VERSIÓN 2: FUNCIÓN QUE RETORNA UN DICCIONARIO MODIFICADO

Los parámetros de la función son el número y el diccionario

✓ Se saca el último dígito del número

✓ Se actualiza el diccionario con las ocurrencias del dígito y

✓ Se retorna el diccionario MODIFICADO

Se repite 10 veces este bloque de instrucciones

```
EjemploConteoDigitos_version2.py
1 def contar(num: int, diccionario:dict)->dict:
2
3     #Se saca el último dígito del número
4     digito = num%10
5     #Se actualiza el diccionario con las ocurrencias del dígito
6     if (digito in diccionario):
7         diccionario[digito] += 1
8     else:
9         diccionario[digito] = 1
10    #Se retorna el diccionario actualizado
11    return diccionario
12
13 #PROGRAMA PRINCIPAL
14 #Se pide el número al usuario
15 numero = int(input("Digite el número de 10 cifras: "))
16
17 #Se crea el diccionario
18 conteo = {}
19
20 #1. Se cuenta el último dígito
21 conteo = contar(numero, conteo)
22 #Se recorta el número
23 numero = numero//10
24
25 #2. Se cuenta el último dígito
26 conteo = contar(numero, conteo)
27 #Se recorta el número
28 numero = numero//10
29
30 # Y LO MISMO 8 VECES MÁS ...
```

VERSIÓN 3: FUNCIÓN QUE RETORNA UN ENTERO

Los parámetros de la función son el número y el diccionario

✓ Se saca el último dígito del número

✓ Se actualiza el diccionario con las ocurrencias del dígito y

✓ Se retorna el número recortado

Se repite 10 veces el llamado a la función contar

```
EjemploConteoDigitos_version3.py x
1 def contar(num: int, diccionario: dict) -> int:
2
3     #Se saca el último dígito del número
4     digito = num%10
5     #Se actualiza el diccionario con las ocurrencias del dígito
6     if (digito in diccionario):
7         diccionario[digito] += 1
8     else:
9         diccionario[digito] = 1
10    #Se retorna el número recortado
11    return num//10
12
13 #PROGRAMA PRINCIPAL
14 #Se pide el número al usuario
15 numero = int(input("Digite el número de 10 cifras: "))
16
17 #Se crea el diccionario
18 conteo = {}
19
20 #Se cuenta el último dígito 10 veces
21 numero = contar(numero, conteo)
22 numero = contar(numero, conteo)
23 numero = contar(numero, conteo)
24 numero = contar(numero, conteo)
25 numero = contar(numero, conteo)
26 numero = contar(numero, conteo)
27 numero = contar(numero, conteo)
28 numero = contar(numero, conteo)
29 numero = contar(numero, conteo)
30 numero = contar(numero, conteo)
31
32 print(conteo)
```

VERSIÓN 4: FUNCIÓN QUE RETORNA UN ENTERO, PERO SIN USAR CONDICIONAL

Los parámetros de la función son el número y el diccionario

✓ Se saca el último dígito del número

✓ Se actualiza el diccionario con las ocurrencias del dígito (usando el método get en lugar del if) y

✓ Se retorna el número recortado

Se repite 10 veces el llamado a la función contar

```
EjemploConteoDigitos_version4.py x
1 def contar(num: int, diccionario:dict)->int:
2
3     #Se saca el último dígito del número
4     digito = num%10
5     #Se actualiza el diccionario con las ocurrencias del dígito
6     diccionario[digito] = diccionario.get(digito,0) + 1
7     #Se retorna el número recortado
8     return num//10
9
10 #PROGRAMA PRINCIPAL
11 #Se pide el número al usuario
12 numero = int(input("Digite el número de 10 cifras: "))
13
14 #Se crea el diccionario
15 conteo = {}
16
17 #Se cuenta el último dígito 10 veces
18 numero = contar(numero, conteo)
19 numero = contar(numero, conteo)
20 numero = contar(numero, conteo)
21 numero = contar(numero, conteo)
22 numero = contar(numero, conteo)
23 numero = contar(numero, conteo)
24 numero = contar(numero, conteo)
25 numero = contar(numero, conteo)
26 numero = contar(numero, conteo)
27 numero = contar(numero, conteo)
28
29 print(conteo)
```


ANALICEMOS LA FUNCIÓN CONTAR



Los parámetros de la función son el número y el diccionario

Cuando se pasa un diccionario como parámetro, en realidad lo que se pasa es una **REFERENCIA** al diccionario (NO UNA COPIA) y por eso se puede modificar dentro de la función

```
EjemploConteoDigitos_version3.py x
1 def contar(num: int, diccionario:dict)->int:
2
3     #Se saca el último dígito del número
4     digito = num%10
5     #Se actualiza el diccionario con las ocurrencias del dígito
6     diccionario[digito] = diccionario.get(digito,0) + 1
7     #Se retorna el número recortado
8     return num//10
```