

# Bitácora 13 del Taller de Herramientas Computacionales

Elías Jiménez Cruz, 409085596

24/01/2019

En esta clase se comenzó a ver un tema importante en la programación, que es el tema de la recursión. Básicamente, una función recursiva es aquella que se utiliza a sí misma para definirse. Con esto, se puede realizar un ciclo a manera de while o de for, sólo que de una manera que no recurre a gastar memoria con un comando de ciclo. Ciertamente es un tema difícil de entender al principio, pero aquí hay algunos ejemplos:

```
{# -*- coding: utf-8 -*-  
'''Programe una función que determine si dos listas son iguales.  
Dos listas se consideran iguales si tienen igual longitud y sus elementos  
en cada índice también lo son.'''  
  
def determinaIgualdad(l1, l2):  
    if len(l1) == len(l2):  
        if l1 == [] and l2 == []:  
            return "Si son iguales."  
        else:  
            if l1.pop() == l2.pop():  
                return determinaIgualdad(l1, l2)  
            else:  
                return "No son iguales."  
        else:  
            return "No son iguales."  
    }
```

```
'''Realice un programa recursivo que determine el factorial de un número natural n'''
def recursion(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return n * recursion(n-1)

n = input(''''Dame el valor de n cuyo factorial quieres conocer: ''')
print "El factorial de %d es: " %(n),
print recursion(n)
```

Como puede apreciarse, todo programa de recursión requiere de una "base" el cual es que detiene el proceso. En el programa de factorial, por ejemplo, al ingresar n, el proceso hace que n se multiplique por la siguiente ejecución de la función, que sería n-1, y a su vez n-1 se multiplicaría por n-2 y así sucesivamente hasta llegar a 1. Al llegar a 1, el proceso devolvería la multiplicación de todas las funciones ejecutadas, dando así el factorial.