**Practica 1**

Los objetivos de esta práctica son:

1) Retomar el contacto con el lenguaje Python.

2) Combinar un programa en Python con Excel/gnumeric para analizar dos conjuntos de datos.

Nota: para familiarizarse con el pasaje de argumentos por línea de comando es conveniente revisar el documento "howto-argparse.pdf" (sección 3)

**1)** Escribir un programa que reciba un string por línea de comando y devuelva True si el string es capicúa y False en caso contrario. Sugerencia: transformar el string a una lista, ej: l = list(s). Nota: no se puede llamar al método "reverse" de listas, hay que hacerlo a mano con un ciclo.

**2)** Escribir un programa que reciba dos vectores (listas) de números enteros de la mismas longitud y calcule el producto interno entre ambos.

**3)** La sucesión de Fibonacci (sugerencia: buscarla en Google) es muy interesante, se usa entre otras cosas para modelar algunos procesos biológicos. Se define mediante la siguiente relación de recurrencia:

f(0) = 0

f(1) = 1

f(n) = f(n-1) + f(n-2)

**3.a)** Escribir un programa (fib1.py) que tome por línea de comando un nro. natural n y que imprima fib(n) haciendo el cálculo de manera totalmente recursiva, es decir: cada vez que haya que calcular un termino de la sucesión haga un llamado recursivo.

**3.b)** Escribir otro programa (fib2.py) similar al anterior que tome por línea de comando un nro. natural n y que imprima fib(n) pero que calcule cada término de la sucesión una sóla vez. Sugerencia: usar una lista que en cada posición tenga el valor de la función para cada n.

**3.c)** Basado en los programas anteriores escribir dos programas (fib1\_delta.py y fib2\_delta.py basados en fib1.py y fib2\_delta.py, respectivamente) que tomen por línea de comando un nro. natural n y que calcule el tiempo promedio de ejecución de 100 iteraciones para todos los valores de 0 a n de la sucesión. Es decir: para cada valor entre 0 <= i <= n hay que calcular el tiempo promedio para 100 iteraciones de f(i). Los programas deben imprimir la siguiente imformación:

0,t(0)

1,t(1)

...

n,t(n)

donde t(i) es el tiempo promedio de ejecución para 100 iteraciones de f(i).

**3.d)** Ejecutar los programas del punto c) para n == 20 y redireccionar la salida a dos archivos, ej.: fib1\_delta.py 20 > fib1.txt

**3.e)** Importar los dos archivos de salida del punto d) en Excel (Windows) o en gnumeric (ubuntu) y graficar las dos series de datos y describir brevemente las conclusiones del análsis de los dos gráficos (ej.: son lineales, exponenciales, etc.).

**3.f)** Con alguna de la implementaciones ¿es posible calcular f(999)? ¿Y f(1000)? ¿Por qué?

**3.g)** Hacer una implementación que no sea recursiva y calcular f(100000).