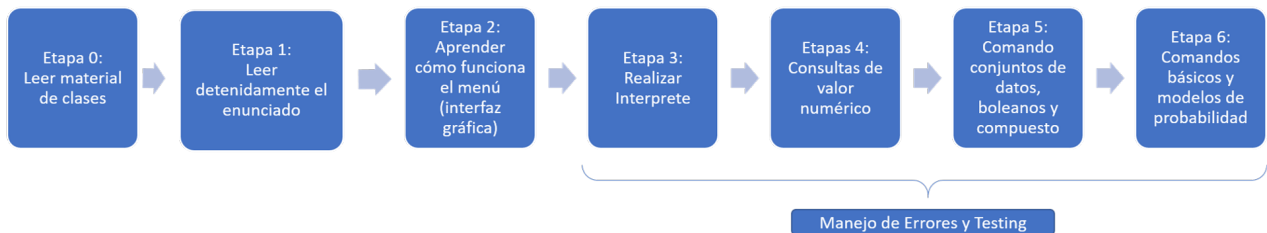




IIC2233 - Programación Avanzada (I/2017)
Workflow - Tarea 3

1. Workflow

A continuación te presentamos unas recomendaciones para que puedas planificar el desarrollo de tu tarea.



■ Etapa 0: Leer material de clases y consejos:

- En primer lugar, si aún no lo has hecho es muy importante leer el material de clases. Para esta tarea debes utilizar principalmente los contenidos de funcional, excepciones y *testing*. Debes tener presente que se espera que utilices los métodos que se entregan en el material para desarrollar las tareas, por ejemplo, que utilices los *built-in* de Python para programación funcional que se presentan en 05-Funcional (*reduce*, *zip*, *lambda*, *map*, *filter*).
- Procura que el resultado de las consultas esté ordenado y sea claro, de tal manera de que sea más sencillo para ti guardar los resultados en `resultados.txt`.
- Tener claro que todo debe estar expresado en funciones lo más atómicas posible, de manera de poder ser reutilizadas. Esto significa que tu programa debe ser modular, es decir que tengas una función por cada tarea que necesites implementar. El siguiente código te muestra un ejemplo para que entiendas mejor el concepto, lo que se busca es poder derivar e integrar funciones, como puedes observar de una función no modular puedes obtener dos funciones modulares.

```
#La siguiente es una función que deriva e integra
def funcion_mala_integrar_derivar(funcion, tipo):
    if tipo == "derivar":
        # Realizar pasos para derivar
        return funcion_derivada
    elif tipo == "integrar":
        # Realizar pasos para integrar
    else:
        return None

#Es mejor que tengan una función para cada tarea que quieran hacer
```

```
def funcion_buena_integrar(funcion):
    #Realizar pasos para integrar
    return funcion_integrada

def funcion_buena_derivar(funcion):
    #Realizar pasos para derivar
    return funcion_derivada
```

- Es recomendable realizar los *test* de las funciones a medida que las crees; de esta manera se te hará más sencillo programarlos y no acumularás trabajo para el último día.

■ Etapa 1: Leer detenidamente el enunciado

- Piensa en una estructura básica para el desarrollo de tu tarea (una lista de las cosas que debes hacer y una idea de cómo podrías hacerlas).
- Procura anotar posibles errores de mal uso, de manera que luego se puedan manejar.

■ Etapa 2: Aprender cómo funciona el menú (interfaz gráfica).

- Te recomendamos estudiar esta sección ya que todos los datos que tu programa debe manejar se obtendrán por la interfaz.
- Es muy útil que aprendas bien cómo funciona, de tal manera que sepas cómo estructurar tus funciones, por ejemplo, para saber el tipo de datos que recibe la función `process_consult` cuando procesas una consulta.

■ Etapa 3: Realizar Interprete

- Desarrolla un intérprete que sea capaz de detectar el tipo de consulta y si esta es una consulta anidada, esto es independiente a si se llega a implementar la funcionalidad. De esta manera te será más sencillo utilizar las distintas funciones que debes implementar.
- Si no te queda mucho tiempo no te detengas en las consultas anidadas y fortalece las consultas simples (consultas no anidadas).

■ Etapas 4: Consultas de valor numérico

- Comenzar por estas consultas te permitirá soltar la mano. Al haber dependencia entre las consultas será más fácil poder visualizar la reutilización de funciones, por lo que una vez que realices las primeras funciones, terminar las otras requerirá menos tiempo.

■ Etapa 5: Comando conjuntos de datos, booleanos y compuesto.

- En cuanto a comandos que retornan un conjunto de datos es muy importante el manejo de estos, te darás cuenta de que si no utilizas la materia de funcional tu programa quedara pegado (debido a la gran cantidad de datos de la base de datos).
- Te recomendamos implementar la consulta Evaluar, una vez hayas implementado los modelos de probabilidad.
- En comando compuesto es útil darse cuenta en qué orden se deberían ejecutar los comandos para ahorrar tiempo (hint: no siempre es necesario ejecutar los tres para obtener el resultado).

■ Etapa 6: Comandos básicos y modelos de probabilidad

- Los comandos básicos en general utilizan resultados de otras consultas, por lo que es recomendable implementarlos luego de tener listos el resto de las consultas.