PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2113 – DISEÑO DETALLADO DE SOFTWARE

Portafolio Personal

Semana 1

Vicente Fuenzalida Marín

Entrega: 11 de agosto de 2017

1. Artefacto por analizar

He decidido observar y comentar acerca de un método creado para una tarea de Introducción a la programación, donde se proponía crear un 'servicio de mensajes encriptados' sencillo en Python.

El método en particular se llama "generar", y tiene por función crear una lista de todos los mensajes ordenados por fecha, almacenados como *strings*. Los mensajes se conforman por: Fecha, Hora, Emisor, Destinatario, Mensaje.

2. Código

```
def generar(enviados,recibidos,mail,clave):
  # crea una lista que contiene todos los mensajes ordenados por fecha, en formato str
  # los mensajes se conforman por Fecha, Hora, Emisor, Destinatario, Mensaje.
  lista_todos=[]
  lista_final=[]
  for i in enviados:
    lista_todos.append([i,"e"])
  for i in recibidos:
    lista_todos.append([i,"r"])
    lista_todos2=ordena(lista_todos)
  for i in range(len(lista_todos2)):
     decript=OTP(lista_todos2[i][0][2],clave)
     if lista_todos2[i][1]=="e":
          lista_final.append(ext_fecha(lista_todos2[i][0])+"  "+ ext_hora(lista_todos2[i][0])+"  De: "+mail+"  Para
          "+lista_todos2[i][0][0]+" - "+binario_a_string(decript))
     elif lista_todos2[i][1]=="r":
          lista_final.append(ext_fecha(lista_todos2[i][0])+"  "+ ext_hora(lista_todos2[i][0])+"  De: "+lista_todos2[i][0][0]+"  Para:
          "+mail+" - "+binario_a_string(decript))
  lista_final.reverse()
  return lista_final
```

3. Reflexión

En primer lugar, se puede ver que existe un comentario al comienzo de la función, en el que se describe en palabras simples el output de la función, así como el formato del mismo. En este punto, cabe notar que no se especifica el formato en términos formales, sino que solo se comenta acerca de la '*semántica*' del mismo (hay una hora, fecha, emisor, etc.). Quizás también sería útil utilizar el idioma inglés para que el código fuese más estandarizado y reutilizable.

Entrando al código mismo, se puede notar que existen varios ciclos '*for*', que podrían ser reemplazados (*refactoring*) por alguna función de tipo *'map'* o utilizando 'list comprehensions'. Más abajo se utiliza la función 'OTP', pero no se señala (mediante comentarios) sobre qué hace o retorna.

Por último, hay una sección donde se "arma" un *string* largo mediante concatenación simple, lo que resulta difícil de entender y mantener. En esta área se puede recurrir al uso de '*string interpolation',* lo que haría que el código quedase más ordenado y legible.

Al ser un método de extensión relativamente corta, y operar sobre tipos de datos similares, se podría decir que existe cohesión, pero al hacer uso de una función de desencriptación externa se aumenta el acoplamiento del método (ya que depende bastante del resultado de otro método externo cuyo comportamiento podría cambiar en el tiempo), y la vez que la cohesión baja. Esto se podría solucionar separando el método 'general' (que estamos estudiando) en dos métodos distintos, cuyas funciones consistirían en que el primero generase una lista, y el segundo desencriptara/formateara los mensajes (labores distintas, que deberían ser desacopladas).