

## TP7

### Ingeniería Reversa, Re-factoría y Re-Ingeniería

- 1) Continuando con el programa getJason.pyc (y su versión fuente getJason.py) del práctico anterior se ha decidido que el programa tiene un código deficiente para lo crítico que resulta para la instalación y por lo tanto se lo someterá a un proceso de **re-factoría**.
  - a) Manteniendo las modificaciones realizadas en el punto anterior analice una intervención que permita aplicar programación orientada a objetos de forma que transforme al programa.
  - b) La transformación consistirá en reorganizar el código para utilizar un patrón de diseño “Singleton”.
  - c) La clase resultante del punto anterior debe poder ser ejecutada desde línea de comandos, al efecto deberán mejorarse las condiciones de chequeo de los parámetros externos de ejecución de forma de hacer a la ejecución más robusta. El objetivo primario es que el programa nunca termine con un error de sistema y siempre lo haga con un error del programa controlado.
  - d) Para hacer la convergencia entre el programa original y su versión re factorizada utilice una estrategia “*Branching by abstraction*”.
  - e) Aproveche a agregar comentarios al programa de su funcionamiento y una carátula donde se especifique es propiedad de la compañía (“copyright UADER-FCyT-IS2©2024 todos los derechos reservados”).
  - f) También mejore el funcionamiento controlando que los argumentos de ejecución sean correctos y en caso de no serlos la terminación sea con un error controlado y no una excepción del lenguaje.
  - g) Agregue además que si el programa se ejecuta con el argumento “-v” emita la versión del mismo. En éste caso lo denominará “versión 1.1”.
  - h) Haga una corrida de revisión con el analizador estático de código pylint y corrija las observaciones hasta que el mismo otorgue un puntaje de 8 o superior.