

# **Trabajo practico numero 1**

Integrantes: Castillo Conrado, Celi Yanina

Materia: Diseño de Algoritmos/Algoritmos II

A continuación, se darán a conocer ciertas características de los programas entregados junto con una descripción de cómo están organizados los archivos, para así saber cómo correr los programas.

Junto con este archive se encuentran dos carpetas, correspondientes a cada ejercicio del proyecto. Dentro de cada carpeta hay dos archivos .java: uno contiene las resoluciones a los incisos y otro los tests para esos incisos. Los nombres de estos archivos y carpetas son bien intuitivos, así que se podrá reconocer a que consigna corresponde cada uno.

Los archivos poseen la documentación necesaria para poder generar javadocs.

En cuanto a los códigos de los programas demandados por las consignas del trabajo, dentro las clases que no son tests se encontrarán con los métodos con la misma estructura y características indicadas por los profesores en el proyecto. Estos métodos son los principales y son los que sobre se ejecutan los tests. También se podrán encontrar con métodos auxiliares que asisten a estos métodos principales. Para ninguna de las clases existe un método main, ya que las pruebas sobre los programas se realizan utilizando las clases que contienen los tests. Las clases test, en cada prueba, poseen comentado el resultado que se espera que se devuelva para ese test y esta nombrado con el método al cual se prueba. Para poder realizar las pruebas se deberán compilar la clase que contiene los métodos de la consigna y la clase que contiene los tests. Luego se deberá ejecutar el compilado de la clase de los tests.

Además de esto querríamos comentar algunas cosas con respecto a los tiempos que nos devolvieron los test ejecutados a los diferentes programas. Pudimos notar que cuando realizábamos pruebas con cadenas cortas, por lo general los tiempos entre fuerza bruta y la otra implementación correspondiente al ejercicio eran similares, incluso las implementaciones que no eran fuerza bruta tardaban más. Pero cuando realizamos tests con cadenas más largas, se notó que los algoritmos de Kmp y divide and conquer son mas eficientes que los de fuerza bruta.