Práctica 1. Programación Funcional en Java

Nuevas Tecnologías de la Programación

UNIVERSIDAD
DE GRANADA

Pedro Serrano Pérez

31 - 03 - 2021

Entorno en el que se ha desarrollado la práctica

El entorno de desarrollo que se ha utilizado para la realización de la práctica ha sido **IntelliJ IDEA**.

Valoración de la práctica

Creo que esta práctica ha sido de gran ayuda para introducir la programación funcional. En general, la práctica me ha resultado muy interesante; conceptos como los flujos, las operaciones sobre ellos, o las funciones lambda, han sido muy novedosos para mi, y trabajar con ellos en la práctica me ha permitido ver la comodidad de su uso en la programación, y la gran potencia que tienen.

En general no ha habido grandes problemas durante el desarrollo de la práctica, aunque el mayor de ellos se ha dado al implementar la heurística de intercambio, debido a que su forma imperativa contenía un bucle while, por lo que me ha resultado algo más difícil de transformar a programación funciona, al tener que que hacer uso de la recursividad.

Cambios realizados para la realización de la práctica

En primer lugar, se han cambiado todas las clases (excepto *Visualizador*) para poder resolver un problema, haciendo uso de cualquiera de las heurísticas, de forma funcional. Para esto, se han conservado los métodos originales, para tener la posibilidad de resolver el problema de forma imperativa, y, para aquellos que poseían bucles, se han añadido nuevos métodos que realizan la misma función mediante programación forma funcional.

Aunque se ha conservado gran parte del diseño base, se han realizado algunos cambios para facilitar la adaptación a programación funcional y la implementación de la heurística del intercambio:

- Se ha añadido el método abstracto *resolverFuncional* a la clase *HeuristicaTSP*.
- Debido a que la heurística del Intercambio hace uso de las rutas generadas por la heurística de Monte Carlo, se ha modificado la clase de esta última heurística para que tenga como atributo un arrayList con las rutas generadas. De esta forma, la clase *HeurísticaIntercambio*, hereda de la clase *HeurisitcaMonteCarlo*, por lo que, en *HeuristicaIntercambio*, se sobreescriben los métodos resolver y *resolverFuncional* de *HeuristicaMontecarlo*.
- Se ha creado un constructor de copia en la clase *Ruta*.