



# Informe Trabajo Práctico Final Parte II

**Materia:** Análisis y Diseño Orientado a Objetos

**Grupo:** Fernando Ares - Diego Tabares

## **Modificaciones al Dominio:**

Para poder agregar las nuevas estadísticas pedidas, debimos analizar cambios que afectarían a tres clases de la solución anteriormente provista y adaptar sus respectivos Tests.

Debajo se detallan las clases modificadas y sus respectivos cambios

## **ManejoDeArchivo:**

En esta clase se recorre línea a línea un archivo csv y se manipulan los datos, para poder guardar el tiempo de uso máximo de una bicicleta y su respectivo Id, debimos agregar lógica que resuelva el problema al momento de recorrer el archivo.

Debido a que esta solución no guarda en memoria todos los datos de un archivo sino que los lee y manipula de acuerdo a lo que necesita procesar, esta clase contiene variables globales que guardan los datos para ser procesados posteriormente, por eso fue necesario agregar nuevas variables globales

```
private float tiempoMaximoRecorrido;
```

```
private List<String> listaIdBicicletaMaximoRecorrido;
```

Esta decisión se tomó para respetar el diseño actual e intentar afectar lo menos posible al resto de las clases que interactúan con esta.

## **ArchivoDeSalidaYML:**

Tuvimos que corregir la lógica para que la salida incluya las nuevas estadísticas, modificar la firma del método de escritura para que acepte nuevos parámetros y agregar un método nuevo de escritura.

### GeneradorDelInforme:

Tuvimos que agregar variables para que esta clase pueda incluir los nuevos datos ya que el diseño actual no utiliza algunos aspectos del DOO, principalmente encontramos que en lugar de guardar un objeto del tipo ManejoDeArchivo, para poder usar sus variables, se declaraban nuevas variables en el GeneradorDelInforme y asignaban valores de ManejoDeArchivo instanciándola en determinados métodos de la clase. De la misma manera que que describimos anteriormente optamos por no cambiar el diseño sino adaptarlo a los nuevos requerimientos.

### **Modificaciones de performance:**

Las mediciones de performance fueron realizadas luego de realizar las modificaciones de dominio.

Para medir la performance utilizamos dos objetos de tipo date, instanciando el primero antes de iniciar el procesamiento del archivo y el segundo al finalizar el mismo.

Luego, los restamos para obtener la diferencia en milisegundos, obteniendo así lo que tardó en procesar cada archivo.

Lo primero que notamos al correr el programa reiteradas veces es que los tiempos de ejecución variaban, con lo que decidimos ejecutarlo diez veces y tomar un promedio para tomarlo como un valor de referencia antes de realizar las modificaciones para intentar mejorar la performance.

### Estos fueron los resultados:

	#1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	#9	# 10	Promedio
Sin Cambios de mejora de Performance	1949	1972	1931	1932	1937	1924	1965	1931	1909	1941	<b>1939.1</b>

Pudimos observar una muy buena performance comparando contra nuestra solución entregada en la parte I, la cual rondaba los 13 segundos.

Una vez obtenido el valor promedio de 1939.1 ms nos dispusimos a pensar cómo podíamos mejorar la performance, y decidimos revisar el informe de PMD en búsqueda de problemas de complejidad ciclomática alta, y encontramos el siguiente:

8 aydoo/ManejoDeArchivo 66 The method 'llenarDatos' has a Cyclomatic Complexity of 13.

Luego de realizar modificaciones a ese método, y volver a generar el reporte de PMD, ese problema ya no estaba listado dentro de la lista de problemas, con lo que nos dispusimos a volver a medir el tiempo de ejecución:

	# 1	# 2	# 3	# 4	#5	# 6	# 7	# 8	# 9	# 10	Promedio
Sin Cambios de mejora de Performance	1949	1972	1931	1932	1937	1924	1965	1931	1909	1941	1939.1
Con Cambios de mejora de Performance	1914	1829	1847	1826	1875	1882	1857	1843	1860	1850	1858.3

Como se puede observar, hubo una mejora en promedio de 80.8 ms luego del cambio realizado.

### ¿Qué aprendimos?

Comprender y realizar cambios sobre código ajeno tiene una complejidad alta. También aprendimos en la parte de performance que en algunos casos, no pudimos reducir la complejidad ciclomática, con lo que los cambios que introdujimos fueron descartados finalmente.