UNIVERSIDAD CATÓLICA BOLIVIANA "SAN PABLO"

UNIDAD ACADÉMICA REGIONAL COCHABAMBA

Departamento de Ingeniería y Ciencias Exactas

Carrera de Ingeniería de Sistemas

Estudio empírico sobre el uso de la librería *Java Stream Collections*

Perfil de Tesis de Licenciatura en Ingeniería de Sistemas

Joshua Abraham Nostas Valdivia

Cochabamba – Bolivia

Abril 2019

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

En la actualidad el uso de la librería *Java Stream Collections* es bastante usada entre los desarrolladores, esta librería proporciona la opción de programar de forma híbrida entre programación orientada a objetos y programación funcional. Permitiendo así realizar operaciones de orden superior en colecciones de objetos, por ejemplo, buscar un elemento en una colección, agrupar los elementos o filtrarlos por un criterio dado. Sin embargo, a pesar de las varias ventajas que puede tener el uso de estas librerías, poco se sabe de los efectos que esta puede tener al rendimiento de los programas y a la utilización de recursos.

Esta tesis de grado propone analizar el uso de esta librería en más de diez mil proyectos java almacenados en github. Este estudio empírico pretende revelar resultados del qué y cómo se utilizan las operaciones proporcionadas por *Java Stream Collections*. Se espera que los resultados sirvan de guía para que los usuarios puedan hacer un mejor uso de la librería, del mismo modo, guiar a los desarrolladores de lenguajes y herramientas de software para dar un mejor soporte a la librería en base a como esta se usa en la práctica.

2. ANTECEDENTES

En la actualidad existen más de un millón de proyectos desarrollados en el lenguaje de programación *JAVA*, solo en *GitHub* están registrados más de 611.678 repositorios desarrollados con este lenguaje. *JAVA* es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue creado con la intención de que los desarrolladores escriban el programa una vez y este pueda ser ejecutado en cualquier dispositivo, esto se conoce como: "*WORA*" lo que significa: "*Write once, Run Anywhere"* [5][6].

JAVA fue creado el año 1991 como una herramienta de programación, y en el año 1996 fue publicada la primera versión de *JAVA* como lenguaje de programación bajo el nombre de JDK (Java Development Kit) 1.0. Desde entonces el lenguaje sufrió múltiples actualizaciones. Uno de los cambios más grandes fue implementado en la versión Java SE 8 publicada el año 2014, en esta versión se introdujo el manejo de expresiones lambda junto con la librería *Stream Collections*, esto aumentó las opciones para los desarrolladores.

2.1. Colecciones *Stream*

Los diseñadores de la interfaz API de Java han incorporado en su actualización una nueva abstracción denominada *Stream*, que permite procesar datos de modo declarativo. Más aún, los *streams* permiten aprovechar las arquitecturas de núcleos múltiples sin necesidad de programar líneas de código multiproceso. Por ejemplo, considere el siguiente código desarrollado de forma tradicional con for-loops donde se quiere sacar el promedio de los estudiantes que tienen una nota mayor a 51.

Figura 1

Ejemplo de función usando un for-loop

```
public int promedio(Estudiante[] estudiantes){
   int sumaNotas = 0;
   int estudiantesNotaMayor51 = 0;
   for(estudiante :: estudiantes){
         if(estudiante.nota() > 51){
               sumaNotas += estudiante.nota();
               estudiantesNotaMayor51 ++;
         }
   }
return sumaNotas/estudiantesNotaMayor51;
}
```

Fuente: Elaboración propia 2019

1. BIBLIOGRAFÍA

[1] Khatchadourian, R. T., Tang, Y., Bagherzadeh, M. y Ahmed, S. (2018) A Tool for Optimizing Java 8 Stream Software via Automated Refactoring. *City University of New York (CUNY) CUNY Academic Works*

[2] Khatchadourian, R. T., Tang, Y., Bagherzadeh, M. y Ahmed, S. (2018) Poster: Towards safe refactoring for intelligent parallelization of Java 8 streams. *City University of New York (CUNY) CUNY Academic Works*

[3]Langer, Angelika (2015) Java performance tutorial - How fast are the Java 8 streams?. *JAX Magazine* https://jaxenter.com/java-performance-tutorial-how-fast-are-the-java-8-streams-118830.html