

Martín Amado 19020
Andrea Amaya 19357
Brandon Hernández 19376

Entrega Fase 1

Investigación Corta Sobre Lisp

Lisp (List Processing) es un conjunto de lenguajes de programación simples que utiliza una notación matemática basada en el cálculo lambda. Los programas desarrollados con Lisp permiten manejar el código como una estructura de datos.

Al tratar de un lenguaje sencillo incrementa la productividad y originalidad. Este cuenta de una memoria de datos y un diccionario de términos. La memoria de datos se utiliza a través de un intérprete, esta es capaz de almacenar distintos tipos de datos haciendo uso de una única posición. Cuando un dato ya no es referenciado, se manda a la recolección de basura para marcar como disponible la memoria que ocupa ese dato. Cada dato es visto como un átomo Lisp (números, cadenas de caracteres y símbolos) y estos son valores, no variables. Siendo así, que en vez de sobrescribir un valor, se hace uso de un nuevo espacio en la memoria.

El intérprete ejecuta estructuras de datos o átomos que se encuentren presentes en la memoria de datos. Este evalúa el primer lugar, donde si es un símbolo se busca en el diccionario de símbolos y se extrae su valor de función de la memoria de datos, en cambio, si es un átomo únicamente se obtiene su valor.

(Marchesi, Menéndez, 2015; Velasco, 2011)

Java Collection Framework

Es una arquitectura unificada que posee la funcionalidad de representar y manipular colecciones, lo que permite administrar las colecciones sin la necesidad de conocer los detalles de la implementación que poseen.

Algunas ventajas de utilizar esta colección son las siguientes:

- Reduce el desgaste de programar.
- Incrementa el desempeño del programador.
- Permite la operabilidad entre programas con distinto API sin relación.
- Reduce esfuerzo para aprender a utilizar los API's.
- Reduce el esfuerzo para implementar y diseñar los API's.
- Fomenta la reutilización de código.

Se divide en tres grandes ramas:

1. Colección de Interfaces; que están divididos en dos grupos:
 - a. `java.util.Collection`
 - b. `java.util.Map`

En donde hay algunas interfaces en las que no se apoyan varias especificaciones por lo que debe de ser especificado en el contrato. Además de que algunas implementaciones restringen la utilización de algunos tipos de datos.

2. Colección de Implementaciones

El propósito general de las implementaciones están resumidas en la siguiente tabla:

Interface	Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Deque		ArrayDeque		LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

Estás admiten todas las operaciones opcionales de la interfaz y no poseen restricciones de datos como ellas. Todas las nuevas implementaciones tienen iteradores que detectan modificaciones inválidas.

3. Colecciones Más Utilizadas

Las aplicaciones que utilizan colecciones de más de un hilo deben de programarse cuidadosamente, la cual se llama programación concurrente. Por lo que Java provee soporte con respecto a este tema, con Java Concurrency Utilities. Las colecciones son muy utilizadas por las interfaces amigables y que las implementaciones se incluyen en las API.

Java Collection Framework que se Utilizaran

Se hará uso de Map. Debido a que para definir funciones y definir variables, se encuentra más eficaz realizarlo por medio de un diccionario para llevar control de todas las definiciones.

Programas Pequeños Lisp

```
(defun factorial (x) "Calcular Factorial" (cond ((= x 1) 1) (t (* x (factorial(- x 1))))))
```

```
(defun FTOC(x) "Farenheit a Celcius" ( * ( - ( * x ) 32 ) 5/9 ))
```

Literatura Citada

1. Marchesi, J., Menéndez, D. (13 de junio de 2015). Lisp, un Lenguaje que se Aprende en 10 Minutos. Extraído de: view-source:<http://www.davidam.com/docu/un-lenguaje-en-diez-minutos.html>
2. Velasco, J. (25 de octubre de 2011). Historia de la tecnología: Lisp. Extraído de: <https://hipertextual.com/2011/10/historia-de-la-tecnologia-lisp>
3. Java. "Collections Framework Overview". Extraído de: <https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>