

UNIVERSIDAD DEL VALLE DE GUATEMALA

Algoritmos y Estructuras de datos

Douglas Barrios



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Proyecto #1

Alejando Ortega #18248

Ángela García #22869

Josué Say #22801

GUATEMALA, febrero de 2023

LISP y Programación Funcional

Lisp es un lenguaje de programación de alto nivel creado en 1958 por John McCarthy. Fue uno de los primeros lenguajes de programación de la historia y sigue siendo ampliamente utilizado en las áreas de Robótica, Inteligencia Artificial, Procesamiento de lenguaje natural, demostración automática de teoremas, etc.

LISP es caracterizado por su sintaxis en forma de función, abarcando en su totalidad al programa, entre otras características se pueden mencionar que destaca por:

- No posee asignaciones.
- Su principal estructura de control es la recursión.
- Los programas y los datos son equivalentes.
- Su principal estructura de datos es la lista.

(Departamento de Ciencias e Ingeniería de la Computación, 2009).

El Lisp es un lenguaje funcional, utilizando funciones matemáticas para el control de los datos. Pero el elemento fundamental en el Lisp es la lista, analizando a LISP como un término amplio, cada función y/o programa generado por LISP está dado en forma de lista (Arroyo, 2005).

En su comparación con otros paradigmas como POO se puede diferenciar en los enfoques para la resolución de problemas y la estructura del código:

- **Enfoque:** POO se enfoca en la representación de los objetos y sus relaciones, mientras que LISP se enfoca en la manipulación de listas y la evaluación de expresiones.
- **Abstracción:** POO ofrece una abstracción de los objetos en el mundo real, mientras que LISP solo ofrece una abstracción de los datos y la lógica en una estructura de listas.
- **Sintaxis:** POO tiene una sintaxis basada en código imperativo y un enfoque en la encapsulación y la herencia, mientras que LISP tiene una sintaxis basada en notación polaca inversa y un enfoque en la evaluación de expresiones.
- **Flexibilidad:** LISP es conocido por su flexibilidad y capacidad para manipular y transformar fácilmente la estructura de datos, mientras que OOP ofrece una abstracción más rígida y un enfoque en la organización y el encapsulamiento de los datos.

(Sánchez, 2017).

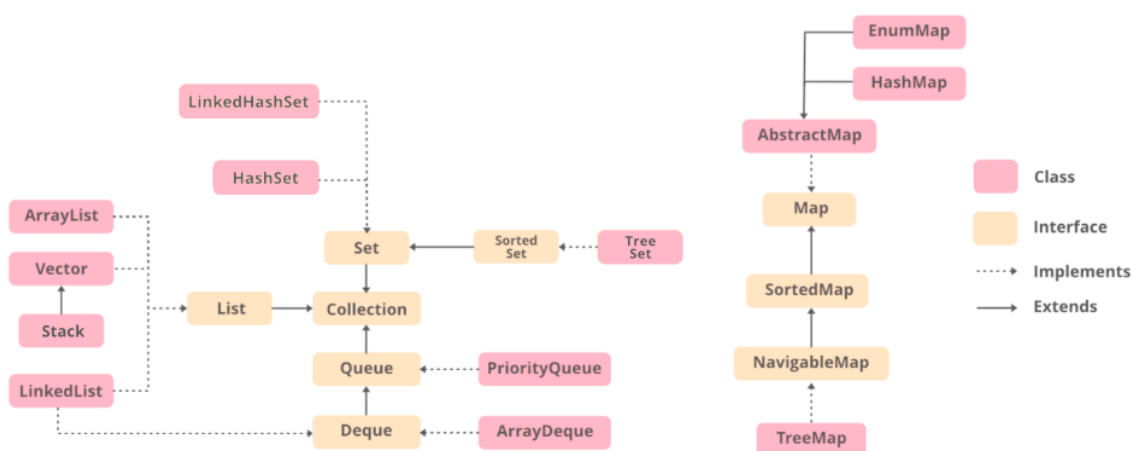
En resumen, LISP es un lenguaje de programación antiguo que ofrece muchas ventajas, sin embargo, en comparación con otros paradigmas de computación, tienen fortalezas y debilidades que van a diferir por el propósito que se busca y las habilidades del programador.

Java Collection Framework

El Java Collections Framework es un conjunto de clases e interfaces que proporcionan la arquitectura para almacenar y manipular un grupo de objetos aplicando los siguientes conceptos (Oracle, s.f.):

- **Interfaces:** tipos de datos abstractos para manipular los detalles de una implementación utilizando abstracción (mivinntec, 2019).
- **Clases:** estructuras de datos implementadas en un programa, estas heredas los métodos y definiciones de una interfaz de colección (mivinntec, 2019).
- **Algoritmos:** métodos y/o conjuntos de declaraciones con el objetivo de ayudar al desarrollador a la clasificación y procesamiento de datos recuperados. Estos actúan de forma polimórfica para utilizar los métodos para adaptar diferentes formas en la implementación de la interfaz de colección de Java (mivinntec, 2019).

Estas interfaces proporcionan una manera estándar de acceder y manipular colecciones, lo que facilita la escritura de código reutilizable y compatible entre diferentes implementaciones de colecciones (mivinntec, 2019).



(HashSet contre TreeSet en Java – StackLima, 2022)

Figura 1. Relaciones entre interfaces de Java Collections Framework.

Indicación de las estructuras del Java Collections Framework utilizadas

El programa a desarrollar debe permitir ejecutar programas sencillos en LISP. Tomar en cuenta que las instrucciones a las que el intérprete debe adaptarse deben ser:

- Operaciones aritméticas.
- Instrucción QUOTE o '.
- Definición de funciones (DEFUN).
- Asignación múltiple (SETQ).
- Predicados (ATOM, LIST, EQUAL, <, >),
- Condicionales (COND), no necesariamente IF.
- Envío de parámetros.

Por lo que en un intérprete de Java a LISP, puede ser útil utilizar varias estructuras de Java Collections Framework (JCF) para almacenar y manipular los datos y la información que se procesa como:

- **Stack:** utilizado para mantener un registro de los elementos a procesar y para llevar a cabo operaciones de evaluación y control de la ejecución.
- **List:** utilizado para almacenar listas de elementos, manipular y procesar los datos en formato de lista que se vayan leyendo en una instrucción de LISP.
- **Map:** esencial para el almacenamiento de valores que requieren un identificador, como en el uso de variables y/o funciones y su identificación en una instrucción de LISP.

(Álvarez, 2022).

Por lo cual se piensa usar estas estructuras de datos mencionadas en el proyecto.

Repositorio de GitHub

[Repositorio](#)

Referencias

- Álvarez, C. (22 de enero de 2022). *Java Collections Framework y su estructura*.
<https://www.arquitecturajava.com/java-collections-framework-y-su-estructura/>
- Arroyo, D. (2005). *Un lenguaje de casi 50 años LISP (I)*.
<https://www.davidam.com/docu/lisp/lisp1.pdf>
- Departamento de Ciencias e Ingeniería de la Computación. (2009). *Programación Funcional en LISP*.
<http://www.cs.uns.edu.ar/~grs/Logica/011.Programacion%20Funcional%20en%20LISP.Color.pdf>
- HashSet contre TreeSet en Java – StackLima. (2022, julio 5).
<https://stacklima.com/hashset-vs-treeset-en-java/>
- mivinntec. (10 de julio de 2019). *Java Collections Framework*. mvit-InnovacionTecnológica.
<https://mvitinnovaciontecnologica.wordpress.com/2019/07/10/java-collections/>
- Oracle. (s.f.). Collections Framework Overview. Recuperado de
<https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>
- Sánchez, D. (24 de noviembre de 2017). *Cual es la diferencia entre programación funcional y orientada a objetos*.
<https://yosoydani.com/la-diferencia-programacion-funcional-orientada-objetos/#:~:text=En%20la%20programaci%C3%B3n%20orientada%20a%20objetos%2C%20se%20almacenan%20los%20datos.como%20una%20transformaci%C3%B3n%20de%20datos.>