

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Colegio Universitario



**Proyecto #1**

Jose Andre Estrada Contreras - 23659

Angel Esteban Esquit Hernández - 23221

Javier Eduardo España Pacheco - 23361

Eduardo Alejandro Rivera Morales - 23281

Algoritmos y Estructuras de datos

Guatemala, 2023

# Lisp

## Investigación

Lisp (List Processing) es un lenguaje de programación diseñado por John McCarthy en 1958, siendo este mismo uno de los lenguajes más antiguos utilizados hasta hoy en día. Fue creado para la investigación en inteligencia artificial en el MIT. Destacó por ser uno de los primeros lenguajes de programación de alto nivel, siendo conocido por su énfasis en la manipulación de listas y su capacidad para tratar código como datos.

Sus principales características son:

- *Sintaxis basada en listas*: Lisp utiliza una sintaxis basada en listas, y el código y los datos comparten la misma representación.
- *Tipado dinámico*: Lisp es un lenguaje de tipado dinámico, lo que significa que las variables pueden contener valores de cualquier tipo.
- *Recolección de basura*: Lisp cuenta con un sistema de recolección de basura que gestiona automáticamente la liberación de memoria no utilizada.
- *Programación funcional*: Lisp es conocido por su enfoque en la programación funcional, lo que implica el uso de funciones como ciudadanos de primera clase.

Empleo:

- Desarrollo de software de inteligencia artificial y machine learning.
- Procesamiento de lenguaje natural.
- Desarrollo de sistemas expertos.
- Investigación en computación simbólica y matemáticas.

Programación Funcional en Lisp:

- Las funciones son entidades que pueden ser asignadas a variables, pasadas como argumentos a otras funciones y retornadas como resultados de funciones.
- Favorece la recursión y el estilo declarativo, definiendo funciones en términos de lo que hacen, enfocándose en la intención más que en los pasos específicos.
- Las funciones pueden ser pasadas como argumentos, devueltas como resultados y almacenadas en variables, proporcionando una composición funcional fluida.

Característica	Programación Funcional	Programación Orientada a Objetos (POO)
Paradigma	Funcional	Orientado a Objetos
Enfoque Principal	Funciones como ciudadanos de primera clase	Organización alrededor de objetos y clases
Manipulación de Datos	Inmutable, énfasis en la	Datos encapsulados en

	inmutabilidad de los datos	objetos, manipulación mediante métodos
Estado y Efectos Secundarios	Minimización de estados mutables y efectos secundarios	Uso de estados y efectos secundarios comunes, especialmente en métodos
Herencia	Menos énfasis, enfoque en composición	Característica central, jerarquía de clases
Polimorfismo	Polimorfismo de funciones (ad-hoc y paramétrico)	Polimorfismo de clases y interfaces
Encapsulamiento	Menos énfasis, datos y código más accesibles	Fuerte énfasis, datos encapsulados en objetos
Ejemplos de Lenguajes	Haskell, Lisp, Scala, Clojure	Java, C++, Python, Ruby, C#

## Java Collections Framework

### Investigación

Java Collections Framework es una biblioteca fundamental en el ecosistema de Java, en la que hay clases e interfaces con las cuales se implementan estructuras de datos y algoritmos comunes para trabajar con ellas. Fue creado en Java 2 en el año 1998 y ha sido renovado con el pasar del tiempo para ofrecer y satisfacer las demandas de los desarrolladores de software actuales. La idea de Java era colocarlo como un nuevo estándar en la manipulación de colecciones de objetos en el lenguaje de Java, lo que permite a los desarrolladores centrarse en la lógica de negocio en lugar de tener que implementar las estructuras de datos desde cero. A su vez, mejora la eficiencia en general del desarrollo del software, ya que esta biblioteca ya fue probada y optimizada para reducir la cantidad de errores que puedan surgir. En el proyecto se utilizarán las interfaces Collection, List, Map, junto con las implementaciones ArrayList, LinkedList y HashMap, para almacenar y manipular colecciones de elementos en distintos contextos, proporcionando opciones para acceso rápido, inserciones eficientes y mapeo clave-valor.

Sus principales características son:

- *Estructura de datos basadas en colecciones:* Esta ofrece una gran variedad de interfaces y clases de uso común, como listas, mapas, colas, etc.
- *Uso de genéricos:* Las clases e interfaces están diseñadas con el uso de genéricos, lo que permite tener una mayor flexibilidad y seguridad para manipular los datos.

- *Iteradores*: Se proporcionan para permitir recorrer las colecciones de manera uniforme de la implementación.
- *Algoritmos de utilidad*: Son una serie de algoritmos útiles para operar en colecciones, como ordenamiento, búsqueda, manipulación, etc.

Empleo:

- Apps empresariales
- Apps Web
- Apps móviles
- Apps de procesamiento de datos