

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC2003 - Algoritmos y estructuras de datos.

Sección 10



Proyecto. Entrega 1.

Oscar Escribá, 21474

Sebastián García 22291

GUATEMALA, 13 de Febrero de 2024

Lisp

Historia

En 1958, John McCarthy desarrolló el lenguaje de programación Lisp, publicando su diseño en 1960 en un artículo titulado "Funciones recursivas de expresiones simbólicas y su cómputo por la máquina, parte I" en Communications of the ACM. Este lenguaje se convirtió en el primer lenguaje de inteligencia artificial, introduciendo conceptos fundamentales como la recursividad, procesos por listas, estructuras de datos en forma de árbol, manejo de almacenamiento automático y tipos dinámicos de datos. Además, McCarthy fue el creador de las sentencias IF, THEN y ELSE, que hoy son comunes en la mayoría de los lenguajes de programación. Lisp permitía la codificación de programas de manera más abstracta, describiendo el problema y el resultado deseado sin necesidad de especificar paso a paso la secuencia de acciones para la ejecución en la máquina. Aunque McCarthy no publicó una segunda parte de su artículo, el legado de Lisp ha influido significativamente en el desarrollo de la informática y la programación.

(Velasco, 2015)

Características de lisp

La principal característica de LISP es su habilidad de expresar algoritmos recursivos que manipulan estructuras de datos dinámicos. En LISP existen dos tipos básicos de datos, los átomos y las listas. Todas las estructuras definidas posteriormente son basadas en estos. Lisp (List-Processing), porque fue creado principalmente para el procesamiento de listas. Es un lenguaje funcional que se apoya en la utilización de funciones matemáticas para el control de los datos. Pero el elemento fundamental en el Lisp es la lista.

A continuación se detallan algunas de las características de las principales variantes del lenguaje LISP:

LISP posee un manejo de memoria automático que libera el espacio utilizado por los objetos que dejan de ser necesarios. Incluye un mecanismo bastante simple para utilizar evaluación perezosa de expresiones. LISP no posee un sistema de tipos estáticos como puede ocurrir en C/C++. LISP asocia los tipos a los valores en vez que a las variables, por ello los errores de mal uso de tipos sólo puedan ser detectados en tiempo de ejecución y no de compilación. Las implementaciones en LISP suelen programarse de manera que cálculos iterativos puedan realizarse en un espacio constante (en memoria) aunque hayan sido descritos mediante el uso de la recursividad.

(EcuRed, s.f)

Donde se utiliza:

Se utiliza para proyectos de código cerrado principalmente para aplicaciones de escritorio específicas de la industria, como el diseño de aviones u otra cosa. Lisp fue creado originalmente como una notación matemática práctica para los programas de computadora, basada en el cálculo lambda de Alonzo Church. Se convirtió rápidamente en el lenguaje de programación favorito en la investigación de la inteligencia artificial (AI). Lisp no tardó en convertirse en el lenguaje favorito en el mundo de la Inteligencia Artificial. Su uso en el MIT

se volvió algo habitual en numerosos proyectos de investigación. A diferencia de otros lenguajes, su base matemática le permitía resolver operaciones y ser utilizado para probar teoremas.

(Quora,s.f)

Java Collections Framework

Java tiene una plataforma de framework. Según la página de Oracle (s.f) menciona que un framework es un grupo de objetos como las clases que hay incluidas como la clase Vector. La colección de framework lo que hace es manipular y representar framework de formas independientes para los detalles de implementación.

Algunas de las ventajas de los framework son:

- reduce el esfuerzo de programación.
- Aumenta el rendimiento.
- Fomenta la reutilización del software.

La collections de framework consiste en:

- Collection interfaces
- Concurrente implementations
- Algorithms
- Infrastructure
- Array utilities

Las Collections Framework se dividen en dos grupos::

- Interfaz básica java.util.Collection se divide en
 - java.util.Set
 - java.util.SortedSet
 - java.util.NavigableSet
 - java.util.Queue
 - java.util.concurrent.BlockingQueue
 - java.util.concurrent.TransferQueue
 - java.util.deque
 - java.util.concurrent.BlockingDeque
 - Java.util.Map
 - java.util.Stack

Estas no son framework verdaderas, no obstante estas interfaz tienen collection-view operations, las cuales ayudan a manipularlas como Collection.

- Map tiene:
 - java.util.SortedMap
 - java.util.NavigableMap

- java.util.concurrent.ConcurrentMap
- java.util.concurrent.ConcurrentNavigableMap

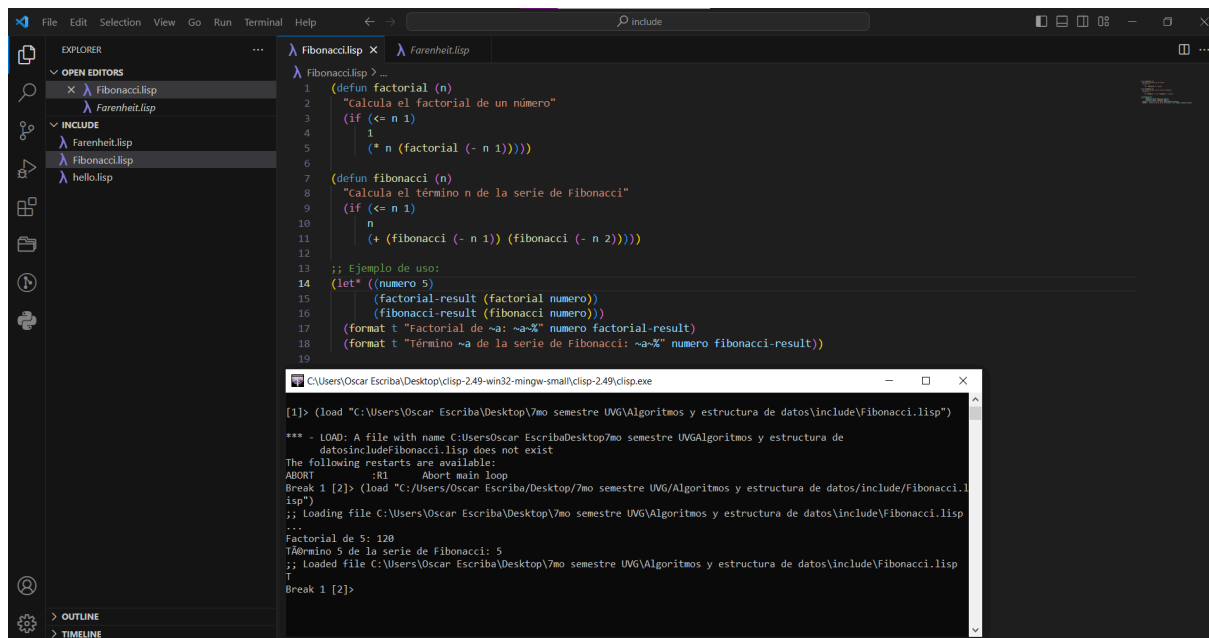
(Oracle, s,f)

Para la realización del intérprete de lisp, se consideró que se necesitara de la utilización de java.util.set con el fin de utilizar HashMap o TreeMap para la implementación del entorno de ejecución. Otra sería java.util.Stack o java.util.Deque para la implementación de pilas y así poder utilizar de mejor forma funciones recursivas. java.util.array para mantener la lista de parámetros de una función o una lista en Lisp.

Programas en Lisp y ambiente de desarrollo:

Link al video: <https://youtu.be/b1x9fh0EssM>

Captura de pantalla, ejecución y código,



The screenshot displays a development environment with a code editor and a terminal window. The code editor shows two Lisp functions: `factorial` and `fibonacci`. The `factorial` function uses a recursive approach with a base case for `n <= 1`. The `fibonacci` function also uses recursion, with a base case for `n <= 1` and a recursive call for `n > 1`. The terminal window shows the execution of the code, including the loading of the files and the output of the functions.

```

1 (defun factorial (n)
2   "Calcula el factorial de un número"
3   (if (<= n 1)
4       1
5       (* n (factorial (- n 1)))))
6
7 (defun fibonacci (n)
8   "Calcula el término n de la serie de Fibonacci"
9   (if (<= n 1)
10       n
11       (+ (fibonacci (- n 1)) (fibonacci (- n 2)))))
12
13 ;; Ejemplo de uso:
14 (let* ((numero 5)
15        (factorial-result (factorial numero))
16        (fibonacci-result (fibonacci numero)))
17   (format t "Factorial de ~a: ~a%" numero factorial-result)
18   (format t "Término ~a de la serie de Fibonacci: ~a%" numero fibonacci-result))
19

```

```

[1]> (load "C:/Users/Oscar Escriba/Desktop/7mo semestre UVG/Algoritmos y estructura de datos/include/Fibonacci.lisp")
*** - LOAD: A file with name C:/Users/Oscar Escriba/Desktop/7mo semestre UVG/Algoritmos y estructura de datos/include/Fibonacci.lisp does not exist
The following restarts are available:
ABORT      :R1 Abort main loop
Break 1 [2]> (load "C:/Users/Oscar Escriba/Desktop/7mo semestre UVG/Algoritmos y estructura de datos/include/Fibonacci.lisp")
;; Loading file C:/Users/Oscar Escriba/Desktop/7mo semestre UVG/Algoritmos y estructura de datos/include/Fibonacci.lisp
...
Factorial de 5: 120
Término 5 de la serie de Fibonacci: 5
;; Loaded file C:/Users/Oscar Escriba/Desktop/7mo semestre UVG/Algoritmos y estructura de datos/include/Fibonacci.lisp
Break 1 [2]>

```

Fibonacci y n factorial

```
File Edit Selection View Go Run Terminal Help
include

EXPLORER
  OPEN EDITORS
    Fibonacci.lisp
    X Farenheit.lisp
  INCLUDE
    Fibonacci.lisp
    Farenheit.lisp
    hello.lisp

Farenheit.lisp
1 (defun celsius-a-fahrenheit (celsius)
2   "convierte grados celsius a grados Fahrenheit"
3   (+ (* celsius 9/5) 32))
4
5 ;; Ejemplo de uso:
6 (let ((temperatura-celsius 25))
7   (format t "Temperatura en Celsius: ~a%" temperatura-celsius)
8   (format t "Temperatura en Fahrenheit: ~a%" (celsius-a-fahrenheit temperatura-celsius)))
9

C:\Users\Oscar Escriba\Desktop\clisp-2.49-win32-mingw-small\clisp-2.49\clisp.exe
The following restarts are available:
ABORT      :R1      Abort main loop
Break 1 [2]: (load "C:/Users/Oscar Escriba/Desktop/7mo semestre UNG/Algoritmos y estructura de datos/include/Fibonacci.lisp")
;; Loading file C:\Users\Oscar Escriba\Desktop\7mo semestre UNG\Algoritmos y estructura de datos\include\Fibonacci.lisp
...
Factorial de 5: 120
Término 5 de la serie de Fibonacci: 5
;; Loading file C:\Users\Oscar Escriba\Desktop\7mo semestre UNG\Algoritmos y estructura de datos\include\Fibonacci.lisp
...
Break 1 [2]: (load "C:/Users/Oscar Escriba/Desktop/7mo semestre UNG/Algoritmos y estructura de datos/include/Farenheit.lisp")
;; Loading file C:\Users\Oscar Escriba\Desktop\7mo semestre UNG\Algoritmos y estructura de datos\include\Farenheit.lisp
...
Temperatura en Celsius: 25
Temperatura en Fahrenheit: 77
;; Loading file C:\Users\Oscar Escriba\Desktop\7mo semestre UNG\Algoritmos y estructura de datos\include\Farenheit.lisp
T
Break 1 [2]:
```

Convertidor de Celsius a Fahrenheit

Bibliografía

EcuRed. (s. f.). LISP - ECURED. <https://www.ecured.cu/LISP>

Oracle (s.f). Collections Framework Overview.
<https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>

¿Cómo se utiliza LISP en la actualidad? (s. f.). Quora.
<https://es.quora.com/C%C3%B3mo-se-utiliza-LISP-en-la-actualidad#:~:text=Se%20utiliza%20para%20proyectos%20de,abierto%20escrito%20en%20lisp%20com%C3%BA>

Tech, T. (s. f.-b). ¿Qué es LISP? ¿Qué Es LISP?
<https://aiofthings.telefonicatech.com/recursos/datapedia/list-procesor>

Velasco, J. (2015, 9 julio). Historia de la tecnología: Lisp. Hipertextual.
<https://hipertextual.com/2011/10/historia-de-la-tecnologia-lisp>