# INSTITUTO TECNOLÓGICO DE COSTA RICA

## TAREA 4 - LISP

*Ariel Herrera*
*Saúl Zamora*

profesor
M. Sc. Saúl Calderón Ramírez

September 25, 2016

## I. Datos históricos

LISP es una familia de lenguajes de programación con una larga y distintiva historia de paréntesis y notación prefija. Originalmente especificado en 1958, LISP es el segundo lenguaje de programación de alto nivel más viejo, sólo superado en edad por FORTRAN, el cual es un año más antiguo. LISP ha cambiado mucho desde sus inicios y muchos dialectos existen hoy en dia; los más conocidos son de propósito general, llamados *Common Lisp* y *Scheme*.

LISP fue originalmente creado como un lenguaje de notación matemática para programas de computación, influenciado por la notación de cálculo lambda de Alonzo Church. Rápidamente fue favorecido como lenguaje para la investigación de la Inteligencia Artificial. Al ser uno de los primeros lenfuajes de programación, LISP fue pionero en muchas áreas de las ciencias de computación tales como árboles, manejo automático del almacenamiento, tipado dinámico, condicionales, funciones de alto orden, recursividad y el compilador hecho con su mismo lenguaje.

El nombre LISP nace de *LISt Processor*, que significa *procesador de listas*. Las listas enlazadas son uno de los mayores tipos de estructuras de datos de LISP y el código fuente de LISP esta hecho de listas. Los programas pueden manipular el código fuente como una estructura de datos, ayudando al programador a crear nuevas sintáxis o lenguajes de dominio específico embebidos en LISP.

## II. Importancia y usos

LISP dió origen a dialectos como *Scheme*, el cual es popularmente utilizado para propósitos educaticos y en la investigación de la inteligencia artificial; el segundo se debe a la facilidad que tiene el código de evaluarse y cambiarse a sí mismo.

## III. Tipos de datos

En LISP, las variables no son tipadas, pero los objetos sí. Los tipos de datos en LISP pueden ser categorizados como:

- Escalares:
  - tipos numéricos
  - caractéres
  - símbolos
- Estructuras de datos:
  - listas
  - vectores
  - vectores de bits (*bit-vectors*)
  - strings

## IV. Expresiones

### A. Operadores aritméticos

+, -, *, /, quotient, gcd (greater common divisor), lcm (least common multiple)

### B. Operadores lógicos

and, nor, not, xor, or

## V. Estructuras de control

LISP originalmente tenía muy pocas estructuras de control, pero muchas más fueron agregadas durante la evolución del lenguaje. Originalmente, el lenguaje solo contaba con el operador *cond*, el cual sería el precursor de la actual estructura de *if-then-else*.

Los programadores de Scheme, usualmente expresan los ciclos usando recursividad de cola. Más recientemente fueron agregadas al lenguaje, estructuras de control con un estilo más imperativo, como el *do* de Scheme y la compleja expresión de *loop* de Common Lisp.

Dada la herencia de procesamiento de listas de LISP, este tiene un largo arreglo de funciones de alto orden relacionadas a la iteracion sobre secuencias. Un ejemplo de esto es la función *map* la cual aplica una función dada sucesivamente sobre una o más listas, recolectando los resultados en una nueva lista.

## VI. Características principales

Una de las principales caracteísticas de LISP es el extenso uso de los paréntesis, dado esto, es nombre LISP logrado ser el acrónimo de *Lots of Irritating Single Parenthesis*.

## VII. Características distintivas

Una de las características distintivas de LISP, es que los programas escritos en este lenguaje, pueden ser capaces de analizarce y reescribirse a sí mismos, lo cual convierte al lenguaje en el ideal para la investigación de la inteligencia artificial.

El lenguaje LISP introdujo el concepto del recolector de basura

## VIII. Ventajas y desventajas

### A. Ventajas

- Manejo de listas nativo.
- Gran poder de expresividad.

### B. Desventajas

- La gran cantidad de paréntesis puede ser abrumadora y difícil de llevar cuenta.

## IX. Ejemplo

```
(defun factorial (n)
   (if (= n 0) 1
       (* n (factorial (- n 1)))))

(defun reverse (list)
  (let ((return-value '()))
    (dolist (e list) (push e return-value))
    return-value))
```

## X. Referencias

## References


[1] Lisp (programming language) (2016). . In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Lisp_(programming_language)

[2] LISP - data types. (2016). Retrieved September 25, 2016, from https://www.tutorialspoint.com/lisp/lisp_data_types.htm

[3] 6.4. Logical operators. Retrieved September 25, 2016, from https://www.cs.cmu.edu/Groups/AI/html/cltl/clm/node75.html

[4] 12.4. Arithmetic operations. Retrieved September 25, 2016, from https://www.cs.cmu.edu/Groups/AI/html/cltl/clm/node125.html