Teoría de la Computación

Programa: Programación Funcional

Miguel Angel Soto Hernandez

**Propósito**

El propósito de este programa es ejecutar funciones básicas mediante programación funcional, es decir, priorizando el uso de la recursividad para llevar a cabo las tareas.

**Manual de uso**

El uso del programa es bastante simple para el usuario debido a que todas las funciones ya se encuentran definidas, y en caso de querer ejecutarlo con otros valores, lo único que se tiene que hacer es cambiar los valores. A continuación se muestra como están estructuradas todas las funciones en el archivo **.lisp** que se encuentra adjunto en la carpeta **.zip** de donde se extrajo este documento.

Ocuparemos como ejemplo la función suma.

En el archivo *programación\_funcional.lisp* se pueden encontrar las siguientes funciones:

* ζ()
* σ(x)
* sum
* mult
* exp
* pred
* monus
* coc
* A (Ackermann)
* ! (factorial)
* fib (fibonacci)

Para cada una de ellas viene un comentario donde se nombra la función que esta representando el código que se encuentra a continuación del comentario, como se puede observar en el código siguiente:

*; Función suma*

(*defun* suma(x y)

    (if (= y 0)

        (progn

            (return-from suma (proyeccion 0 (list x y)))

        )

        (return-from suma (sigma(proyeccion 2 (list x y (suma x (- y 1))))))))

La línea de ejecución se encuentra justo debajo de la función que se esta definiendo, la cual contiene una impresión en pantalla, y se ve de la siguiente manera:

(format t "Función suma de 100 + 50 = ~d ~%"(suma 100 50))

Si se requiere cambiar los valores de ejecución que vienen por defecto, se requiere hacer un par de cambios, los cuales cambiarían su funcionamiento:

Ejemplo:

En esta llamada a la función ejecutaremos la función suma de 100 más 50.

(format t "Función suma de 100 + 50 = ~d ~%"(suma 100 50))

Para cambiar la ejecución a 5 más 5, se tendría que modificar el código para que quede de la siguiente manera

(format t "Función suma de 5 + 5 = ~d ~%"(suma 5 5))

De esta manera ahora el programa a la hora que se ejecute ejecutará la función suma o la que se haya seleccionado con los valores que el usuario introdujo. De esta misma manera se pueden modificar todas las funciones que se encuentran contenidas dentro del programa, exceptuando la función cero ya que esta simplemente regresa un cero como salida.

**Ejemplo de ejecución**

Para ejecutar el código es necesario tener instalado LISP en su computadora, y simplemente ejecutar el archivo *programación\_funcional.lisp* desde la consola con el comando *clisp* *programación\_funcional.lisp* .

De no contar con la instalación de LISP, puede acceder al sitio web <https://www.tutorialspoint.com/execute_lisp_online.php> el cuál es un compilador en línea de LISP. A continuación, puede copiar y pegar el código y ejecutarlo en la casilla de Execute dentro del sitio web.

A continuación se muestra un ejemplo del código ejecutado con todas las funciones antes mencionadas.

