



CENTRO DE CIENCIAS BÁSICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
LENGUAJES INTELIGENTES
5° "A"

ACTIVIDAD 16

Profesor: Alejandro Padilla Díaz

Alumno: Joel Alejandro Espinoza Sánchez

Fecha de Entrega: Aguascalientes, Ags., 16 de noviembre de 2020

Actividad 16

Tenemos cuatro ejercicios propuestos a resolver:

- 1) En el primer ejercicio se nos propone desarrollar un programa en LISP el cual convierta una cantidad de dólar a euros, por lo tanto, se desarrolla el siguiente código.

```
;Convertir una cantidad de dólar a euros
(define (DOLAR-EURO cd)
  (/ (* cd 20) 25))

(DOLAR-EURO 1)
(DOLAR-EURO 10)
(DOLAR-EURO 5)
(DOLAR-EURO 20)
(DOLAR-EURO 22)
(DOLAR-EURO 35)
```

Éste, junto con algunas ejecuciones de ejemplo produce el siguiente resultado al ejecutarse:

```
Bienvenido a DrScheme, versión 299.100.
Lenguaje: Estudiante Principiante.
0.8
8
4
16
17.6
28
> (DOLAR-EURO 8)
6.4
>
```

Observemos que también se realiza una ejecución ejemplo cuando $cd = 8$ para observar un funcionamiento correcto del programa.

- 2) En el segundo ejercicio, trabajamos sencillamente la definición recursiva de cómo calcular el factorial de un número, por lo tanto, se desarrolló el siguiente código junto con algunas ejecuciones de ejemplo:

```
(define fact
  (lambda (n)
    (if (zero? n) 1
        (* n (fact (sub1 n))))))

(fact 20)
(fact 5)
(fact 0)
(fact 70)
```

Este código produce a su vez el siguiente resultado:

```
Bienvenido a DrScheme, versión 299.100.  
Lenguaje: Estudiante Principiante.  
2432902008176640000  
120  
1  
11978571669969891796072783721689098736458938142546425857555362864628009582789845319680000000000000000  
> (fact 18)  
6402373705728000  
>
```

También probamos con otro ejemplo para revisar que trabaja correctamente.

- 3) En el tercer ejemplo, tenemos un ejercicio principal y tres incisos a trabajar:
- En el primer ejercicio de este ejemplo, tratamos de ingresar tres números y el programa los “volteará” como si fuera un conversor de tres dígitos que cambia los números a la inversa, entonces desarrollamos el código:

```
;Definición de la función  
(define (convert3 n1 n2 n3)  
  (+ (* n3 100) (* n2 10) n1))  
  
(convert3 1 1 1)  
(convert3 5 3 4)
```

Observamos que los resultados obtenidos son de la ejecución completa del programa, pero señalamos con rojo aquellos resultados pertenecientes a este ejercicio:

```
Bienvenido a DrScheme, versión 299.100.  
Lenguaje: Estudiante Principiante.  
111  
435  
14  
35  
91  
22  
32.5  
60.5  
1.5  
1.8  
1.8  
>
```

- a. Para los ejemplos posteriores, se pedía determinar el valor en $n = 2$, $n = 5$ y $n = 9$ y expresarlo en forma de programa. Por lo tanto, para este inciso se pedía evaluar:

$$n^2 + 10$$

El código desarrollado es el siguiente:

```
;a) Evaluar n^2 + 10
;Definimos la función
(define (g n)
  (+ (* n n) 10))

(g 2)
(g 5)
(g 9)
```

Los resultados obtenidos nuevamente remarcados son los siguientes:

```
Bienvenido a DrScheme, versión 299.100.
Lenguaje: Estudiante Principiante.
111
435
14
35
91
22
32.5
60.5
1.5
1.8
1.8
>
```

b. Para este inciso se pedía evaluar:

$$\frac{n^2}{2} + 20$$

El código desarrollado es el siguiente:

```
;b) Evaluar (1/2) * n^2 + 20
;Definimos la función
(define (h n)
  (+ (* (/ 1 2) (* n n)) 20))

(h 2)
(h 5)
(h 9)
```

Los resultados generados son los siguientes:

```
Bienvenido a DrScheme, versión 299.100.
Lenguaje: Estudiante Principiante.
111
435
14
35
91
22
32.5
60.5
1.5
1.8
1.8
>
```

c. Para este inciso se pedía evaluar:

$$2 - \frac{1}{n}$$

El código desarrollado es el siguiente:

```
;c) Evaluar 2 - (1/n)
;Definimos la función
(define (b n)
  (- 2 (/ 1 n)))

(b 2)
(b 5)
(b 9)
```

Los resultados son los siguientes:

```
Bienvenido a DrScheme, versión 299.100.
Lenguaje: Estudiante Principiante.
111
435
14
35
91
22
32.5
60.5
1.5
1.8
1.8
>
```

- 4) En el último ejemplo, se pedía hacer un ejemplo donde se sumaran monedas de cierto peso, en este caso de \$1, \$2, \$5 y \$10 en una bolsa. El código es el siguiente:

```
;Definir el programa suma-monedas. Consume cuatro números:
;el número de monedas de $1, $2, $5, $10 en la bolsa y
;produce la cantidad de dinero en la bolsa

;Definición de la función
(define (suma-monedas m1 m2 m3 m4)
  (+ (* m1 1) (* m2 2) (* m3 5) (* m4 10)))

(suma-monedas 100 40 30 20)
(suma-monedas 23 40 200 10)
```

El resultado de la ejecución del código anterior junto con una prueba extra, es el siguiente:

```
Bienvenido a DrScheme, versión 299.100.
Lenguaje: Estudiante Principiante.
530
1203
> (suma-monedas 1 1 1 1)
18
> |
```