



Taller 1: Procesamiento de datos simples. Fundamentos de Programación

Carlos Andres Delgado S, Ing *

Septiembre de 2016

1. Reglas del taller

Las reglas de este taller son:

1. Este taller puede ser presentado en grupos de hasta tres personas.
2. Deben seguir la receta de diseño vista en el curso, en caso de no hacerlo se penalizará en un 30 % la nota del taller.
3. En el taller se dan nombres sugeridos de funciones, por favor usarlos, debido al gran número de estudiantes se usarán pruebas automáticas para la validación de sus soluciones.
4. En el taller se dejan algunos ejemplos, sin embargo, debe crear para caso 3 ejemplos. Esto hace parte de su nota.
5. Deben entregar en el campus virtual, en el enlace provisto por el profesor el archivo **.rkt** o **.scm** generado por Dr Racket. **Enviar la entrega sólo un integrante del grupo** y este debe ser **antes de la fecha del cierre**
6. Dentro de este archivo, a manera de comentarios deben ir los nombres y códigos de los integrantes
7. Se permitirán entregas hasta 4 horas después del cierre, pero estas tendrán una penalización de 10 % de la nota del taller por hora de retraso (se calcula de forma ponderada), por ejemplo un atraso de 30 min, implica una penalización del 5 %. No se aceptan entregas por correo.
8. Si conoce el curso (es repitente o está adelantado en temas) **NO USE RECURSIÓN** para procesar las listas, use **first** o **rest** ya que estas son de tamaño fijo.

2. Enunciado

1. [5 puntos] Implemente la función **dos-rangos**, esta función recibe un número y retorna verdadero si está en alguno de los dos rangos:
 - a) Si es mayor o igual que 10 o menor o igual que 20
 - b) Si es mayor o igual 100 y menor o igual que 120

*carlos.andres.delgado@correounivalle.edu.co

```
(dos-rangos 10)
;;Retorna #t
(dos-rangos 21)
;;Retorna #f
(dos-rangos 25)
;;Retorna #f
(dos-rangos 120)
;;Retorna #t
```

2. [10 puntos] Escriba una función **calcula-funcion** para calcular la siguiente expresión:

$$f(x, y, z, w) = \frac{x^2 + y * z - \frac{z}{4}}{x + y * (z + w) * \frac{y^2}{z^3 + 3 * (x + 2)}} \quad (1)$$

```
(calculaFuncion 6 2 3 15)
4.675
(calculaFuncion -1 4 3 -9)
-0.887681159420289855072463
(calculaFuncion -1 -9 3 -9)
-0.1847375690607734806629834...
```

3. [10 puntos] Escriba una función **calcular-compra** que recibe una lista de productos, una lista de precios y una lista de cantidades de tamaño 5 cada una. El precio total por producto obedece la siguiente regla:

- Si se compran menos de 5 elementos se cobra el 100 % del valor del producto
- Si se compran 5 y menos que 10 elementos se cobra el 80 % del valor del producto
- Si se compran entre 10 y menos que 15 elementos se cobra el 70 % del valor del producto
- Si compra más de 15 elementos se cobra el 60 % del valor del producto

```
(define productos (cons 'arroz (cons 'aguacate (cons 'cerveza (cons 'cd (cons 'audifono empty))
))))
(define precios (cons 1000 (cons 1500 (cons 1800 (cons 500 (cons 3000 empty))))))
(define cantidadesA (cons 1 (cons 3 (cons 10 (cons 13 (cons 1 empty)))))
(define cantidadesB (cons 20 (cons 14 (cons 5 (cons 13 (cons 6 empty)))))

(calcular-compra productos precios cantidadesA)
25650
(calcular-compra productos precios cantidadesB)
52850
```

4. [20 puntos] Construya una función **calcular-venta** que permita calcular el precio total de una venta. A esta función le ingresa una lista con 5 elementos de una estructura conocida como venta-producto que contiene: producto, cantidad, valor unitario y descuento. Para calcular el valor de cada producto debe tener en cuenta su precio unitario, el descuento (se da un número que significa el porcentaje entre 0 y 100) y la cantidad que se compra.

```
(define productoA (make-venta-producto "pollo" 2 3000 20)
(define productoB (make-venta-producto "carne" 1 4000 0)
(define productoC (make-venta-producto "cerveza" 3 2000 30)
(define productoD (make-venta-producto "libro" 10 7000 10)
(define productoE (make-venta-producto "cerdo" 3 8000 5)
(define listaProductos (cons productoA (cons productoB (cons productoC (cons productoD (cons
producto E empty)))))

(calcular-venta listaProductos)
98800
```

5. [25 puntos] Desarrolle una función **calcular-valor-plan**, esta función recibe tres estructuras así:

- a) La primera estructura **info-hotel**, contiene información acerca del hotel de hospedaje, este contiene 3 campos
- 1) Nombre hotel
 - 2) Número de noches
 - 3) Costo por noche
 - 4) Tipo habitación **info-habitacion**, esta es una estructura que contiene:
 a' Número camas
 b' Numero baños
 c' Número duchas

El costo total de la habitación es el número de noches por el costo por noche. Si la habitación tiene más 2 camas o más de 3 baños o más de 2 duchas, se aplica un cargo adicional del 25 % a este valor

- b) La siguiente estructura **info-transporte**, la información del transporte
- 1) Ciudad origen
 - 2) Ciudad destino
 - 3) Valor ticket
 - 4) Numero personas
 - 5) Otros del viaje **otros-viaje-info**: Que es una estructura que contiene:
 a' Valor servicio transporte al aeropuerto
 b' Valor servicio ancianos o niños

El costo total del transporte, es valor del ticket por el número de personas más el valor de los servicios adicionales (transporte al aeropuerto y niños/ancianos). La empresa ofrece un descuento del 40 % si hay más de 5 personas en el plan.

- c) La última estructura **info-guia**, contiene el plan de guía por la ciudad de destino de la siguiente forma:
- 1) Ciudad
 - 2) Nombre del plan
 - 3) Número de sitios
 - 4) Valor por sitio

El valor total de la guía en la ciudad de destino, es el número de sitios por el valor del sitio. Si se desea visitar más de 4 sitios se aplica un cargo del 20 %.

La función que usted diseñe debe retornar una estructura **info-plan** la cual contiene los siguientes datos

- a) Valor hotel
- b) Valor transporte
- c) Valor plan de guía
- d) Valor total

```

(define habitacion (make-info-habitacion 2 4 5))

(define hotel
  (make-info-hotel
    "Tulua City"
    2
    20000
    habitacion
  )
)

(define otrosViaje
  (otros-viaje-info
    4000
    6000
  )
)

(define transporte
  (make-info-transporte
    "Cali"
    "Cartagena"
    120000
    6
    otrosViaje
  )
)

(define guia
  (make-info-guia
    "Cartagena"
    "Plan Estafa Fundamentos Programación"
    5
    40000
  )
)

(calcular-valor-plan hotel transporte guia)
;;Debe retornar
(make-info-plan
  50000
  438000
  240000
  728000
)

```

6. [10 puntos] Usando el punto anterior, diseñe la función **desplegar-informacion** que reciba la respuesta de la función (**debería almacenarse en alguna variable**) y el nombre de la persona, como salida se tiene un texto con la información del plan, este no entiende de estructuras :').

```

(define salidaPlan (calcular-valor-plan hotel transporte guia))
(define desplegar-informacion plan "Carlos")
;;Debe retornar "Señor Carlos su plan tiene un costo total de 728000, en hotel son 50000, en
transporte son 438000 y su plan de visita 240000, gracias por creer en nosotros :)"

```

Para presentar la información puede apoyarse en las funciones **string-append** y **number->string**:

```

(string-append "El valor de a es: " (number->string 3) " y su cuadrado es: " (number->string (
  expt 3 2)))
;;Debe retornar "El valor de a es: 3 y su cuadrado es: 9"

```

7. [20 puntos] Construya una función **construir-figura** que permita construir figuras en el Canvas de DrRacket, esta recibe una lista de estructuras **figuras** (la lista de es de tamaño 4) con los siguientes campos:

- Tipo: Puede ser "Cuadrado" o "Circulo"
- Color: Puede ser "Rojo", "Amarillo", "Verde" o "Azul"
- Tamaño: En el caso del cuadrado es el Lado y en el circulo es el Radio
- Centro de la figura: Es una estructura **posn** (disponible en Dr Racket), tenga presente que el centro del cuadrado es la posición $(\frac{L}{2}, \frac{L}{2})$

El tamaño del canvas es 400 por 400, por lo que tenga presente las figuras no pueden salirse.

```
(define figuraA (make-figura "Cuadrado" "Azul" 50 (make-posn 50 50)))  
(define figuraB (make-figura "Cuadrado" "Rojo" 100 (make-posn 100 150)))  
(define figuraC (make-figura "Circulo" "Verde" 50 (make-posn 200 50)))  
(define figuraD (make-figura "Circulo" "Amarillo" 100 (make-posn 300 300)))  
  
(define lista-figuras (cons figuraA (cons figuraB (cons figuraC (cons figuraD empty)))))  
  
(construir-figura lista-figuras)
```

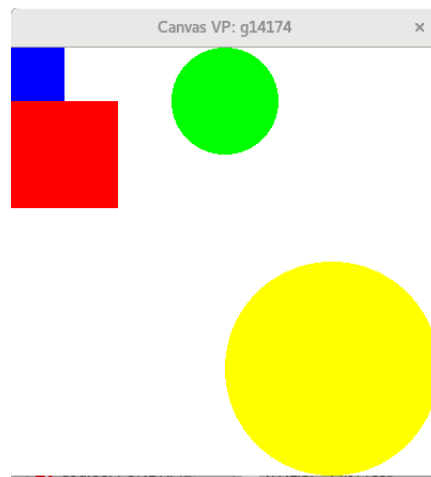


Figura 1: Resultado

Para este caso apóyese en la función **string=?** y recuerde agregar el paquete educativo **draw.rkt**