

Taller 2 – Programación Declarativa

Indicaciones Generales

- Tiempo máximo: 2 horas.
- Los ejercicios deben resolverse individualmente en un entorno Racket.
- En caso de utilizar WhatsApp, redes sociales u otro medio de comunicación con compañeros del aula, se asignará una nota de 2.
- Cada ejercicio indica su ponderaje correspondiente; la suma total equivale al 100 % de la nota del taller.

Ejercicio 1 – Contar elementos positivos en una lista

Objetivo: Usar filter y length para determinar cuántos elementos positivos contiene una lista.

Entrada: Entrada: '(3 -2 7 0 -5 9).

Salida esperada: Salida esperada: 3 elementos positivos.

Ponderaje: 5 %

Ejercicio 2 – Generar lista de cuadrados pares

Objetivo: Practicar map y filter aplicados secuencialmente.

Entrada: Entrada: '(1 2 3 4 5 6 7 8).

Salida esperada: Salida esperada: '(4 16 36 64).

Ponderaje: 5 %

Ejercicio 3 – Calcular el factorial de un número

Objetivo: Implementar recursión simple en Racket.

Entrada: Entrada: n = 5.

Salida esperada: Salida esperada: 120.

Ponderaje: 5 %

Ejercicio 4 – Elevar cada número al cubo

Objetivo: Aplicar funciones lambda dentro de map.

Entrada: Entrada: '(2 3 4).

Salida esperada: Salida esperada: '(8 27 64).

Ponderaje: 5 %

Ejercicio 5 – Sumar todos los elementos impares

Objetivo: Usar filter y foldl combinados para obtener una suma condicional.

Entrada: Entrada: '(1 2 3 4 5 6 7).

Salida esperada: Salida esperada: 16.

Ponderaje: 5 %

Ejercicio 6 – Determinar si una lista contiene números negativos

Objetivo: Usar any oormap con una función lambda para verificar condiciones.

Entrada: Entrada: '(5 9 -3 2).

Salida esperada: Salida esperada: #t (contiene negativos).

Ponderaje: 8 %

Ejercicio 7 – Calcular la suma acumulada de una lista

Objetivo: Reforzar el uso de foldl para crear acumuladores personalizados.

Entrada: Entrada: '(1 2 3 4).

Salida esperada: Salida esperada: '(1 3 6 10).

Ponderaje: 8 %

Ejercicio 8 – Concatenar cadenas de texto en una lista

Objetivo: Practicar foldl con operaciones sobre cadenas.

Entrada: Entrada: '("Hola" " " "Mundo").

Salida esperada: Salida esperada: "Hola Mundo".

Ponderaje: 8 %

Ejercicio 9 - Generar lista con el doble de los números mayores que 5

Objetivo: Combinar map y filter con condiciones lógicas.

Entrada: Entrada: '(3 6 8 2 10).

Salida esperada: Salida esperada: '(12 16 20).

Ponderaje: 8 %

Ejercicio 10 - Invertir el orden de una lista

Objetivo: Usar foldl o recursión para invertir el orden de los elementos.

Entrada: Entrada: '(1 2 3 4).

Salida esperada: Salida esperada: '(4 3 2 1).

Ponderaje: 8 %

Ejercicio 11 - Crear una función que reciba una función como parámetro

Objetivo: Comprender la naturaleza de las funciones de orden superior.

Entrada: Entrada: Función cuadrado y lista '(1 2 3 4).

Salida esperada: Salida esperada: '(1 4 9 16).

Ponderaje: 10 %

Ejercicio 12 - Reto integrador: combinar múltiples funciones

Objetivo: Usar map, filter y foldl en un mismo programa para calcular el promedio de los números mayores a 5 en una lista.

Entrada: Entrada: '(3 8 10 4 9 2 7).

Salida esperada: Salida esperada: 8.5.

Ponderaje: 25 %