**TC2006 – Lenguajes de Programación***Funciones de Orden Superior en Haskell*

Matrícula1\_\_\_\_\_\_\_\_\_ Matrícula2\_\_\_\_\_\_\_\_\_\_ Matrícula3\_\_\_\_\_\_\_\_\_\_

**IMPORTANTE**: Solo se evaluarán y asignarán estrellas a los   
miembros de los grupos creados para el ejercicio

1. Programar la función sin recursión explícita **resto\_col** que elimine la primera columna de una matriz dada como una lista de renglones.

Probar con:

> resto\_col [[1,2,3],[4,5,6]] **=> [[2,3],[5,6]]**

1. Programar la función sin recursión explícita **sumaMatriz** que obtenga la sumatoria de todos los elementos de una matriz.

Probar con:

> sumaMatriz [[1,2],[3,4]] **=> 10**

1. Programar la función sin recursión explícita **trans** que obtenga la transpuesta de una matriz.

Probar con:

> trans [“ab”,”cd”] **=> [“ac”, “bd”]**

> trans [[1,2,3],[4,5,6]] **=> [[1,4],[2,5],[3,6]]**

1. Programar la función **aplicaN** que genere procedimientos que apliquen n veces sobre un valor, la función dada como parámetro de entrada.

Probar con:

> (aplicaN **(+ 1) 7) 5 => 12**

> (aplicaN (\x->x\*x) 2) 3 **=> 81**

> (aplicaN tail 3) [1,2,3,4] **=> [4]**

1. Programar la función sin recursión explícita **secuencia** que genere listas con los números del 1 al N, utilizando la FOS **until**.

Probar con:

> secuencia 3 **=> [1,2,3]**

> secuencia 5 **=> [1,2,3,4,5]**