**TC2006 – Lenguajes de Programación***Programación Secuencial en Erlang*

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

1. Programar la función **mayor** en Erlang que reciba 3 argumentos numéricos y regrese el mayor.

Probar con:

> ej1:mayor(7,-5,3). **=> 7**

> ej1:mayor(1,2,3). **=> 3**

> ej1:mayor(9,9,9). **=> 9**

1. Programar la función **suma** en Erlang que calcule la sumatoria:

Probar con:

> ej1:suma(0). **=> -1**

> ej1:suma(3). **=> -1+1+3+5 = 8**

1. Programar la función recursiva **negativos** en Erlang que regrese una lista con los valores negativos de una lista de números.

Probar con:

> ej1:negativos([3,3,3]). **=> []**

> ej1:negativos([2,-3,4,-5]). **=> [-3,-5]**

> ej1:negativos([-9,-2,5,3,-2]). **=> [-9,-2,-2]**

1. Programar la función de orden superior **filtra** en Erlang que trabaje igual que la FOS lists:filter (sin utilizarla).

Probar con:

> ej1:filtra(fun(X) -> X > 1 end, [-1,2,-3]). **=> [2]**

> ej1:filtra(fun(X) -> X < 0 end, [-1,2,-3]). **=> [-1,-3]**

1. Programar la función **impares** que sin utilizar recursividad explícita elimine todos los elementos que no sean impares de una lista de sublistas.

Probar con:

> ej1:impares([[1,2,3],[4,5,6]]). **=> [[1,3],[5]]**

> ej1:impares([[2,2],[2,2],[2,2]]). **=>[[],[],[]]**