**TC2006 – Lenguajes de Programación***Evaluación Perezosa y Otras Facilidades en Haskell*

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

1. Programar la función **digpares** en Haskell usando **guardias** que cuente la cantidad de dígitos pares de un entero no negativo.

Probar con:

> digpares 5 => 0

> digpares 32465 => 2

1. Programar la función **parimpar** en Haskell usando **comprensión de listas** que separe una lista de enteros en dos listas que contengan los valores pares en una e impares en otra.

Probar con:

> parimpar [1,2,3,4,5] => ([2,4],[1,3,5])

> parimpar [4,8,3,9] => ([4,8],[3,9])

1. Programar las funciones **rango\_where** y **rango\_let** usando guardias y **where** para la primera y usando el **let** para la segunda, donde a partir de N valores en una lista, determine el rango de los valores definido por el valor menor y el valor mayor.

Probar con:

> rango\_where [95] => (95,95)

> rango\_let [75.3,60.4] => (60.4,75.3)

> rango\_where [100,89,60] => (60,100)

> rango\_let “hola” => (‘a’,’l’)

1. Programar la función **mclaurin** en Haskell usando listas infinitas que calcule la siguiente función a partir de N términos de la siguiente serie infinita de Mclaurin:

Probar con:

> mclaurin 0.5 1 => 1.0  
> mclaurin 0.5 2 => 1.5

> mclaurin 0.5 4 => 1.875