**TC2006 – Lenguajes de Programación***Programación Recursiva en Racket*

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

1. Programar en Racket la función **armonica** que calcule la suma de n términos de la serie:

Probar con:

> (armonica 1) **=> 1**

> (armonica 3) **=> 1.8333333**

1. Programar en Racket el procedimiento **bajasube** que despliega los números enteros del N al 1 al N, para N > 0, separados por espacios.

Probar con:

> (bajasube 3) **=> 3 2 1 1 2 3**

> (bajasube 5) **=> 5 4 3 2 1 1 2 3 4 5**

1. Programar en Racket la función **fibo** que regrese el n-ésimo elemento de la serie de Fibonacci: 1,1,2,3,5,8,13,21,… donde cada elemento después de los primeros 2 se calcula sumando sus 2 predecesores.

Probar con:

> (fibo 2) **=> 1**

> (fibo 7) **=> 13**

1. Obtén la versión terminal de la función fibo como **fibot**. En este caso requerirás de 2 argumentos adicionales en la función auxiliar. Pueden comparar tiempos de ejecución con la función (time <llamada>).

Probar con:

> (fibot 2) **=> 1**

> (fibot 7) **=> 13**