

**Grado en Diseño y Desarrollo de Videojuegos**  
**Algoritmos para Juegos**  
**Práctica 2: Recordando cómo programar con una estrategia recursiva**

1.- Diseña una función recursiva que calcule el número de dígitos que tiene un número entero en base 10.

2.- Diseña una función recursiva que sume los dígitos de un número. Por ejemplo, si la entrada es 9999 el resultado que devolverá es 36.

3.- Implementa una función recursiva, `sumaParImparRec(num)`, que extraiga las cifras de un entero positivo, sume las pares, reste las impares y devuelva el resultado. Por ejemplo, si el número es 5634, el cálculo que debe hacer es:  $\text{sumaParImparRec}(5634) = -5 + 6 - 3 + 4 = 2$ .

4.- Diseña una función recursiva que calcule la suma de los  $n$  primeros números naturales.

5.- Diseña una función recursiva que permita calcular la suma de la siguiente serie:

$$\sum_{i=1}^n \frac{1}{i}$$

donde  $n \geq 1$  es un número entero que se recibe como argumento de la función.

6.- Diseña un procedimiento recursivo que imprima la lista de números naturales comprendidos entre dos valores *ini* y *fin* dados por el usuario.

7.- Diseña una función recursiva que calcule  $x^y$  mediante multiplicaciones sucesivas, siendo  $x$  e  $y$  dos números enteros.

8.- Diseña una función recursiva que calcule  $x*y$  mediante sumas sucesivas, siendo  $x$  e  $y$  dos números enteros.

9.- Definir un subprograma recursivo, denominado `MultiplosRec` que reciba dos números `tope` y `divisor`, de tipo entero, tal que `tope > divisor` y que imprima por pantalla los múltiplos de `divisor` menores que `tope`. Ejemplo: `MultiplosRec(6,31)` imprimirá 6 12 18 24 30 (en líneas diferentes).

10.- Puedes calcular recursivamente los números combinatorios sabiendo que, para  $n \geq m$ ,

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$$

y que

$$\binom{n}{n} = \binom{n}{0} = 1$$

Diseña un programa que, a partir de valores  $n$  y  $m$  leídos por teclado (con  $m$  entre 0 y  $n$ ), muestre el valor de  $\binom{n}{m}$ . Para ello, el programa llamará a una función de nombre *combinaciones* definida recursivamente.

11.-- Diseña una función recursiva que reciba una cadena de caracteres y devuelva la cadena inversa.