

# Actividad 1. Repaso de recursividad

## Objetivo

Repasar los conocimientos adquiridos en el curso pasado sobre programación en JAVA y recursividad. Introducir el uso de JUnit para realizar pruebas unitarias.

## Procedimiento

- 1.- Repasa los apuntes de recursividad de primer curso.
- 2.- Crea una clase Java que implemente los ejercicios que se proponen en esta actividad; un método por ejercicio.
- 3.- Crea una clase Java que contenga el método main(), para probar el correcto funcionamiento de los métodos implementados en la clase anterior.
- 4.- Implementa las pruebas con JUnit. Para saber cómo crear de forma sencilla las pruebas con JUnit y NetBeans, consulta alguno de los siguientes enlaces:

<https://asaes.wordpress.com/2009/07/05/junit-uso-de-asserts-en-netbeans-6-5-1/>

<http://diagonalasterisco.blogspot.com.es/2009/07/integracion-de-junit-y-netbeans.html>

<http://alcasoft.blogspot.com.es/2013/05/java-pruebas-unitarias-con-junit-y.html>

## Evaluación

Estos contenidos serán evaluados indirectamente en las pruebas que se realizarán durante todo el cuatrimestre.

## Tiempo estimado

4 horas

## Ejercicios

1. Implementa un método recursivo que calcule el factorial de un número. Por ejemplo  $4! = 24$ .
2. Implementa un método recursivo que calcule el cuadrado de un número  $n$ , sabiendo que  $n^2 = (n-1)^2 + 2n - 1$ .

3. Implementa un método recursivo que sume los dígitos que componen un número entero positivo que se pasa como parámetro. Es decir, dado el número 1369, debe devolver el entero 19.
4. Implementa un método recursivo que calcule el máximo común divisor de dos números enteros, aplicando el algoritmo de Euclides. Dicho algoritmo consiste en ir restando el número más pequeño del más grande hasta que queden dos números iguales, que serán el máximo común divisor de los dos números.
5. Implementa un método recursivo que devuelva si una cadena es palíndromo.
6. Implementa un método recursivo que sume los elementos de un array de enteros.
7. Implementa un método recursivo que invierta los números de un array de enteros.
8. Implementa un algoritmo recursivo que devuelva el menor elemento de un array de enteros.
9. Implementa la búsqueda binaria en un array de números reales con un método recursivo. El método devuelve la posición del número a buscar en el array. Si no está, devuelve -1.
10. Haciendo uso de la clase Stack<E> de Java, implementa un método recursivo que copie una pila genérica. Los elementos de la pila destino deben quedar en el mismo orden que en la pila original.