

# Actividad 1. Repaso de recursividad

## Objetivo

Repasar los conocimientos adquiridos en el curso pasado sobre programación en JAVA y recursividad, haciendo uso del entorno NetBeans. Repasar el uso de JUnit para realizar pruebas unitarias.

## Procedimiento

- 1.- Repasa los apuntes de recursividad de AEDI (tienes un video disponible en Moovi Tema 1/ Sección 1.1 / Actividades Grupo Reducido / *AEDI - Video de repaso de recursividad*).
- 2.- Implementa los ejercicios que se proponen en esta actividad. Se recomienda implementarlos en una sola clase Java, escribiendo un método por ejercicio, para facilitar las pruebas posteriores con JUnit.
- 3.- Para probar el correcto funcionamiento de los métodos implementados en la clase anterior crea una clase Java que contenga el método main().
- 4.- También puedes probar el correcto funcionamiento de los métodos implementados haciendo uso de JUnit. Para saber cómo implementar pruebas unitarias sencillas con Junit en Netbeans puedes revisar los siguientes enlaces:  
Pruebas unitarias con Junit 4 en NetBeans: <https://www.youtube.com/watch?v=dKW1FFgTa84>
- 5.- Además, para facilitar las pruebas unitarias de esta actividad y que podáis probar vuestros métodos sin necesidad de crear los tests, se os proporciona un test básico en Moovi, Tema 1/ Sección 1.1/ Actividades Grupo Reducido / *Fichero de test: SolActividad1Test.java*. Una explicación de cómo usar los test en Netbeans la tenéis en el video disponible en esa misma sección (*Video para uso de ficheros de test en Netbeans*).

## Evaluación

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

## Tiempo estimado

2 horas presenciales

## Ejercicios

1. Implementa un método recursivo que calcule el factorial de un número entero positivo. Por ejemplo,  $4! = 24$ .
2. Implementa un método recursivo que calcule el cuadrado de un número  $n$ , siendo  $n \geq 0$ , 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 invierta una palabra de forma recursiva.
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 ordenado de forma creciente, 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.