



Objetivos

- Conocer e implementar los principales algoritmos, basados en el esquema divide y vencerás, para ordenación y búsqueda.
- Conocer y reutilizar la librería del JDK. Las clases **Arrays** y **Collections**.

Requisitos

- Seguir el esquema de nombrado de paquetes **org.mp.sesion0x**. En este paquete, en la carpeta de fuentes **src**, se crearán todos los programas que se proponen en la sesión, dándoles un nombre alusivo a lo que realiza el programa y que se indica en cada ejercicio en negrita.
- Las pruebas JUnit, en el mismo paquete **org.mp.sesion05**, en la carpeta de fuentes **test**.
- Al final de la sesión, el alumno deberá cargar el trabajo realizado a su repositorio indicando la clave correspondiente a la sesión.
- Documentar todas las clases.
- Las sesiones se han diseñado para cubrir una semana de trabajo.

Ejercicios propuestos

1. Escriba la clase **SolucionesBusqueda**, que resuelve la búsqueda binaria recursiva. En esta solución utilizaremos la excepción **ElementoNoEncontrado** para avisar de que una búsqueda fracasó.
2. Implemente los algoritmos **inserción**, **mergesort** y **sort** que resuelven la ordenación por comparación de un vector de N elementos.
 - Utilice para las comparaciones objetos de tipo **Comparable**. Cree una clase **Ordenacion** con métodos estáticos, uno para cada algoritmo de ordenación.
 - Para el tercer algoritmo **sort**, deberá utilizar el método **sort** de la clase **Arrays** del paquete **java.util**

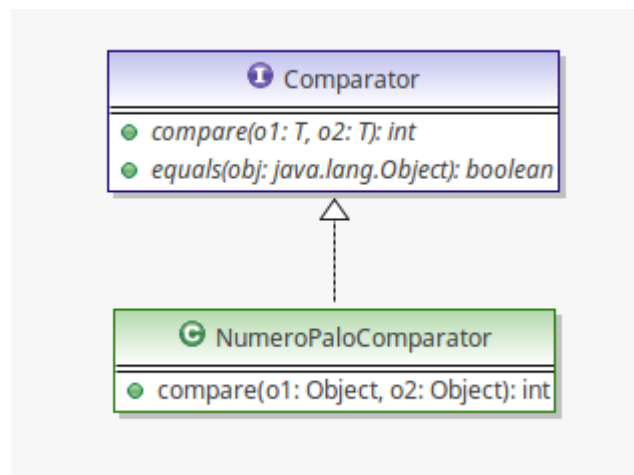
G Ordenacion	
●	<u>ordenacionPorInsercion(a: Comparable[]): void</u>
■	<u>ordenacionPorInsercion(a: Comparable[], izq: int, der: int): void</u>
●	<u>mergeSort(a: Comparable[]): void</u>
■	<u>mergeSort(a: Comparable[], vectorTemp: Comparable[], izq: int, der: int): void</u>
■	<u>mezclar(a: Comparable[], vectorAux: Comparable[], posIzq: int, posDer: int, posFin: int): void</u>
●	<u>sort(a: Comparable[]): void</u>

Trabajo autónomo

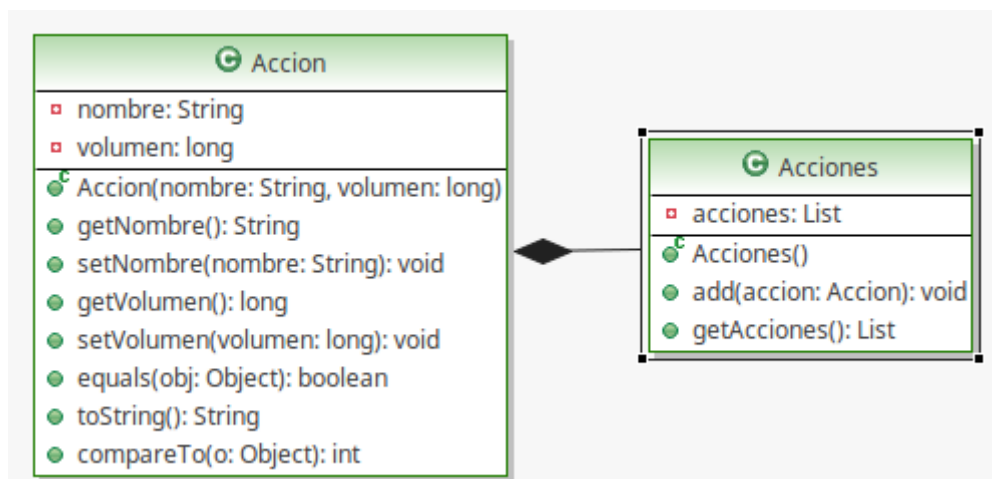
3. A partir del ejercicio de la sesión 01 (El ejercicio de la Baraja). Haga una copia de las clases creadas y documentadas al un nuevo paquete org.mp.sesion06.

Ordenar la baraja en base a nuevo orden: primero número y dentro del mismo número los palos {O,C,E,B}, es decir, O 1, C 1, E 1, B 1, O 2, C 2, E 2, B 2 O 12,C 12,E 12,B 12. Para ello, deberá implementar la clase **NumeroPaloComparator** que implementa la interface java.util.**Comparator**.

En el paquete de test para este ejercicio deberá situar los juegos de pruebas **BarajaTestA** y **BarajaTestB**.



4. A partir del diagrama siguiente, implemente las clases: **Accion**, **Acciones** y **VolumenComparator**.



5. El alumno deberá estudiar la API de las siguientes clases: java.util.**Arrays** y java.util.**Collections**