# Mergesort

## Introducción

Dentro de los métodos de ordenamiento, se encuentran los ordenamientos por comparaciones. Mergesort es uno de ellos y es uno de los mas eficaces O(nlogn) lo cual es un costo muy bueno para el tipo de ordenamiento (el mejor caso para los métodos de comparaciones es O(nLogn)).

## Análisis del algoritmo

Luego de saber que hace Mergesort, lo siguiente es saber como lo hace y para ello esta el siguiente algoritmo:

**MergeSort(arr[], l, r)**

If r > l

**1.** Find the middle point to divide the array into two halves:

middle m = (l+r)/2

**2.** Call mergeSort for first half:

Call mergeSort(arr, l, m)

**3.** Call mergeSort for second half:

Call mergeSort(arr, m+1, r)

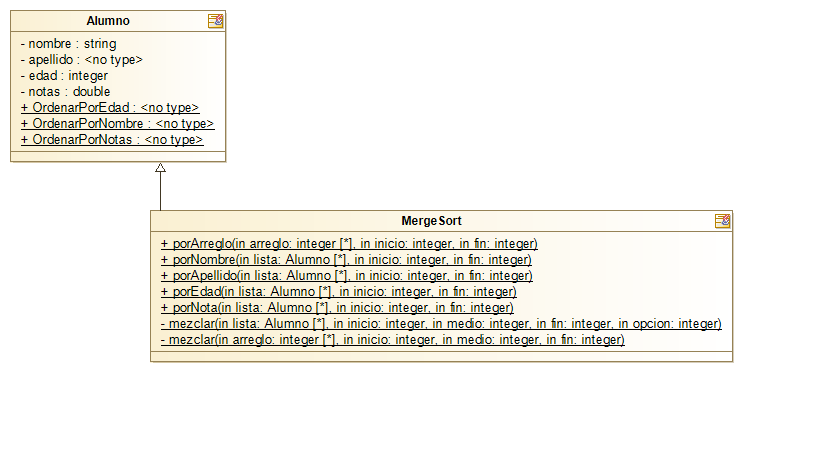
**4.** Merge the two halves sorted in step 2 and 3:

Call merge(arr, l, m, r)

Como vemos, Mergesort realiza llamadas recursivas para llevar a cabo el método de ordenamiento el cual particiona el arreglo a la mitad recursivamente de izquierda a derecha para luego ordenar los sub arreglos.

## Diseño del algoritmo

Para el diseño del algoritmo se elaboro el siguiente UML:



En la cual se utiliza una clase “Alumno” implementada por Gonzalo Villalobos de la cual sus instancias serán ordenadas dentro de una lista de tipo ArrayList. El algoritmo puede ordenar por diferentes criterios (los cuales son los atributos de la clase Alumno) según lo que requiera el jefe de proyecto.