

En cuanto al contenido del texto, he de decir que toca de forma tangencial un tema relativamente importante y esencial, que es la importancia de la disyuntiva entre la memoria persistente y no persistente. Esto se debe de entender como del eterno debate entre una memoria rápida y una duradera, lo cual inevitablemente obliga a la creación de dos clases diferentes de almacenamiento de memoria. Trayendo este debate al tema que nos ocupa, la ArrayList sería el almacenamiento rápido y un recurso de almacenamiento (como una base de datos SQLite) sería el almacenamiento duradero.

Por otro lado tenemos el uso de las clases para representar datos complejos, lo cual es algo fundamental ya que nos otorga la facilidad de agrupar datos estrechamente relacionados y sus correspondientes funciones para operarlos. La importancia que se le otorgó en su momento a este concepto de “agrupamiento y encapsulación de parámetros y métodos” es lo que dió origen a la POO y, por lo tanto, hace que sea fundamental tener en mente, de forma casi inconsciente, cuando se programa en lenguajes como Java esto mismo, que no es sino la principal razón de ser de la POO.

Si rescatamos los dos conceptos anteriormente mencionados, en la tarea dada en el aula se usa una clase para [ordenar](#) los datos relacionados que queremos guardar y sus métodos correspondientes (un registro en nuestro caso), mientras que la ArrayList (una estructura de datos) se utiliza para agrupar dichos registros de forma temporal antes de decidir si se almacenan de forma persistente en la base de datos o si se quedan en una forma volátil.

IMPORTANTE: para **ordenar**, no **estructurar** (seguramente vendría bien dejar claro el uso de las estructuras de datos respecto del de las clases, aunque seguramente este sea un tema complejo para el nivel actual del aula).