# GUIÓN CLASE

**1.-** La explicación en clase de esta versión, comenzará con el estudio de los distintos métodos de la clase AccesoBD.

**2.-**Se hará un dibujo para ver el ciclo de vida de la gestión de una base de datos(SQL embebido) desde un programa.

**3.-** Los elementos que forman parte en esta gestión son: **conexión, comando**, **transacción** y **ejecución de sentencias** (**ejecutarConsulta()** para sentencias **SELECT** y **ejecutarUPDATE()** para las órdenes **INSERT, UPDATE y DELETE**). El nombre ejecutarUPDATE() se puso, cuando este método se codificó en JAVA.

**4.-** Realizar un dibujo del proceso de ejecutarConsulta(). Le pasamos una senctencia SQL y la clase de objetos de la cual queremos montar el ArrayList de salida.

**5.-** Hacer una explicación de los cursores y de DataReader con un dibujo y ver la semejanza con el Resulset de JAVA. Este sería el montaje del cursor(DataReader y Resultset) en modo **conectado**, pero con el fin de generar un arrayList de objetos del tipo pasado como parámetro en la orden ejecutarConsulta(). Explicar el concepto de modo **desconectado,** tanto en Java como NET(cacherowset y dataset). Si cerramos una conexión en el modo conectado, perdemos todos los datos del cursor, por eso es necesario descargarlos(registros) en un ArrayList para hacer que la gestión de los datos sea más genérica e intuitiva.

**Esta es la forma de trabajar del patrón MVC . Haciendo procesos de abstracción de las distintas capas. El objetivo de esta arquitectura, es el viaje de OBJETOS y ARRAYS de objetos entre las distintas capas(en ambos sentidos), CAPA DE PRESENTACIÓN, CAPA de NEGOCIO y PERSISTENCIA(Capa de datos). Si es necesario, RECORDAR dibujo de este movimiento de objetos. Ver cómo el CURSOR se llena con los registros del DataReader (Resultset). Estos objetos al final gestionan trozos de datos que están en memoria.**

**6.-** Se hará un seguimiento del proyecto con la TRAZA para estudiar el ciclo de vida de la aplicación.

**7.-** Explicar el concepto de la clase **Object**. Para hacer métodos con datos genéricos, debemos usar el tipo Object. Si tenemos un array de Object, podremos introducir en dicho array cualquier tipo de objeto(libro, cliente, artículo etc.). Si por el contrario, el array fuese de la clase Libro, solo podríamos introducir objetos libro dentro del array. Otro ejemplo, si tengo un objeto de la clase Object, puede asignarle cualquier tipo de objeto. Para realizar métodos genéricos, necesitamos comprender la clase Object y la reflexión. Este último concepto se explicará, indicando que podremos obtener todos los datos del objeto, interrogándolo mediante reflexión.

**8.-** Se hará un dibujo de cómo van los datos desde la capa de presentación a la base de datos y viceversa.