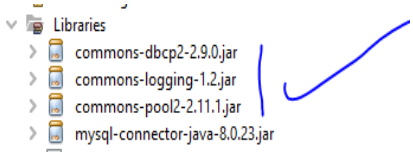
ACCESO A DATOS
UD2: MANEJO DE CONECTORES

EXPLICACIÓN CLASE POOL

- Pre-requisitos para que funcione:

Importar librerias para trabajar con el pool.



Que se añaden a la del driver de la BD.

Y por supuesto indicar la BD y la IP de trabajo ya que varía según a donde conectemos.

- Métodos.

```
public class Pool {

    static Connection Con;
    static BasicDataSource basicdatasource = new BasicDataSource();
    private static final String BD = "ejemplo";//*******Indica la BD **************
    private static final String IP = "192.168.56.101";//*******Indica la IP ***************

    public static Connection IniciaPoolconFichero() throws SQLException {...11 lines }

    public static Connection IniciaPool() throws SQLException {...22 lines }

    public static Connection getConexion() throws SQLException {...7 lines }

    public static Connection getCurrentConexion() throws SQLException {...6 lines }

    public static void Cerrar() {...8 lines }
}
```

IniciaPoolconFicheros: Este método no lo vamos a utilizar pero permite iniciar el pool de conexión con un fichero externo de properties. Este método os lo dejo por si en un proyecto futuro tuvierais que recoger los parámetros de la BD, usuario y password de un fichero externo de configuración.

IniciaPool: Metodo que SI vamos a usar. Este método es IMPORTANTISIMO, ya que inicia el pool de conexiones. No es la conexión, inicia el que podamos conectarnos. Se ejecuta al inicio de la aplicación.

```
public Enunciado4() {
    try {
        Pool.IniciaPool();
        initComponents();
```

Solo con llamarlo ya inicia el pool de conexiones. A partir de ese momento ya podemos conectarnos a la BD.

getConexión: Metodo que SI vamos a usar pero no directamente ya que lo utilizaremos a partir de getCurrentConexion. Este método es que realmente conecta a la BD. Observa que indicamos también que la conexión tiene el autocommit a falso. Esto va a indicar que para poder hacer efectivas las inserciones,borrados y actualizaciónes vamos a tener que realizar un commit en el código.

getCurrentConexion: Metodo que SI vamos a usar MUY IMPORTANTE. Este método lo usaremos para realizar cualquier tipo de operación a la BD. Comprueba si hay Conexión. Si no la hubiera conecta llamando a getConexión y si ya la hubiera la devuelve.

```
ResultSet rs;
try ( Statement sentencia = Pool.getCurrentConexion().createStatement()) {
    rs = sentencia.executeQuery(consulta);
```

Cerrar: Metodo que SI vamos a usar MUY IMPORTANTE. Cierra la conexión.

```
}finally{
    Pool.Cerrar();
}
```

- ¿Cómo trabajaremos?. IMPORTANTE

  El proceso será siempre el mismo. Al iniciar la aplicación iniciamos el pool

  Pool.IniciarPool();

  Cada vez que realicemos un proceso (que no una consulta, ya que un proceso puede requerir muchas consultas) abriremos la conexión con Pool.getCurrentConexion() la primera vez y las posteriores ya utilizará la conexión solicitada.

  Finalmente Cerramos la conexión con Pool.Cerrar();