**Resumen Minerva**

**1-Investigar compatibilidad entre rfid y nfc**

**Si** hay una compatibilidad entre el rfid y el nfc ya que el nfc es el subconjunto con respecto a las funcionalidades del RFID. Debemos pedirle prestado los tags con el alien a la profesora carolina para que lo probemos.. Hay programas que desde nuestro nfc en el Android lee los rfid, tales como:

- NFC TagInfo

- RFID NFC TOOLS

- RFID Tag Finder

Ahora solo debemos comunicarnos desde el android studio a traves de librerias para poder leer los Tags.

**2-Investigar que otras bd se pueden comunicar a android studio**

Entre las base de datos suministradas a Android studio esta el SQLSERVER.

Lo explica muy bien aquí:

<http://moztrodev.blogspot.com/2013/10/hazlo-facil-conectar-aplicacion-android.html>

### Hazlo fácil: Conectar aplicación Android a Base de Datos SQLServer

Quizás debería estar durmiendo a estas horas de la madrugada, o quizás viendo una película o pasando el rato navegando..., quizás, pero no, en lugar de eso estoy aquí picando algo de código y me pasó por la mente "Hey, si de escribir código se trata, porque no hacerlo para un tutorial, al fin y al cabo le debes un tiempo al blog", y me respondí "Cierto, pero antes de pasar a cosas interesantes, empecemos con algo sencillo que nos servirá para calentar motores", así que adelante... (la falta de sueño causa más daño de lo que creí)

[](http://www.androidworld.it/wp-content/uploads/2012/04/android-db.jpg)

Con este tutorial lograrán conectar cualquier aplicación android con una base de datos SQL Server tanto local como remotamente de una manera sencilla, tal y como si se tratara de una aplicación java común y corriente, y esto gracias al driver jTDS de JDBC.

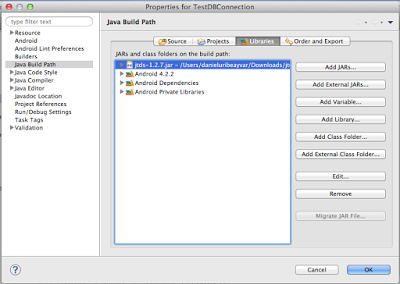
**Descargar el driver jTDS**

Entonces lo primero que haremos será ir  a la [página oficial](http://jtds.sourceforge.net/license.html) y descargarnos el driver. **NOTA**: El driver que utilizaré será el 1.2.7 y android 4.2.

**Creando el proyecto y añadiendo el driver**

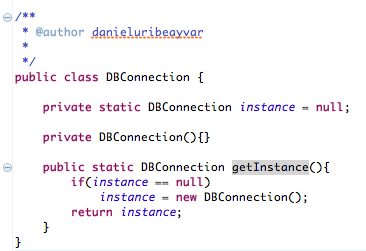
Una vez descargado, abrimos nuestro IDE(en mi caso eclipse) y creamos una nueva aplicación de android.

Una vez creado importaremos el jar del driver jTDS a nuestro proyecto, para esto hacemos click derecho sobre nuestro proyecto->Properties->Java Build Path->Libraries y en Add External JARs..., buscamos nuestro jar descargado y click en OK.

[](http://3.bp.blogspot.com/-7XUfEvGTKA4/UlTdsTXlXJI/AAAAAAAACXg/R6e2qxP6-44/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.37.33.png)

**La clase DBConnection**

Creamos una nueva clase que llamaremos *DBConnection* y la crearé como un [Singleton](http://moztrodev.blogspot.mx/search/label/singleton),

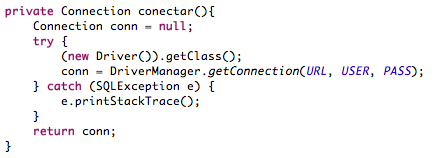
[](http://2.bp.blogspot.com/-Suw5pNrPqfs/UlTbOs-VcvI/AAAAAAAACXM/x4aj62lGWOo/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.26.41.png)

Añadimos los atributos URL, USER y PASS que contendrán la información y credenciales para conectarnos a la base de datos.

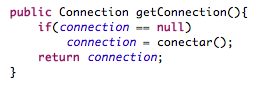
[http://2.bp.blogspot.com/-Vl9A4Pq32r8/UlTgjqisWbI/AAAAAAAACXs/zValkOKtGrw/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.48.22.png](http://2.bp.blogspot.com/-Vl9A4Pq32r8/UlTgjqisWbI/AAAAAAAACXs/zValkOKtGrw/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.48.22.png)

Ya saben lo que pienso sobre el [código duro](http://moztrodev.blogspot.mx/search/label/malas%20practicas) pero dado que es un ejemplo, podemos dejarlo pasar.

Después creamos el método que hará la conexión, y en el instanciamos el driver para después pasarle nuestros datos de conexión al *DriverManager*.

[](http://4.bp.blogspot.com/-UcCKdRaLZww/UlThkOD5miI/AAAAAAAACX0/S6wAeYGXLrU/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.54.12.png)

Bueno, ya para terminar creare otro método singleton que obtendrá la conexión hecha en nuestro método *conectar()*.

[](http://2.bp.blogspot.com/-Z8x0Wxcqt4M/UlTiac5L23I/AAAAAAAACX8/b54SsBqlqWU/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.58.01.png)

Entonces la clase final nos quedaría de la siguiente manera:

[](http://4.bp.blogspot.com/-JGVj1wv_YWc/UlTi91BcGMI/AAAAAAAACYE/lP8VavLMolo/s1600/Captura+de+pantalla+2013-10-08+a+la(s)+23.59.51.png)

**Llamando a nuestra conexión**

Ahora para llamar llamar a nuestra clase lo haremos de la manera como se llaman a las clases Singleton.

[http://4.bp.blogspot.com/-RhDTNP4AOxQ/UlTkI0zcHkI/AAAAAAAACYQ/pjpQnfYhi2E/s1600/Captura+de+pantalla+2013-10-09+a+la(s)+00.05.14.png](http://4.bp.blogspot.com/-RhDTNP4AOxQ/UlTkI0zcHkI/AAAAAAAACYQ/pjpQnfYhi2E/s1600/Captura+de+pantalla+2013-10-09+a+la(s)+00.05.14.png)

**Otorgando permisos a nuestra aplicación**

Por poco lo olvido, necesitarán otorgarle permisos a su aplicación para que pueda hacer uso del internet y conectarse a su base de datos si ésta se encuentra remota. En el *AndroidManifest.xml*de su aplicación añadan la siguiente línea de código.

[http://4.bp.blogspot.com/-SLZRxR3vygM/UlTlHtEiOcI/AAAAAAAACYY/3FuuK2MdruY/s1600/Captura+de+pantalla+2013-10-09+a+la(s)+00.09.30.png](http://4.bp.blogspot.com/-SLZRxR3vygM/UlTlHtEiOcI/AAAAAAAACYY/3FuuK2MdruY/s1600/Captura+de+pantalla+2013-10-09+a+la(s)+00.09.30.png)