

Vamos a echar un vistazo a los distintos JDBC ResultSet Types que existen. Recordemos que el API de JDBC es parte del temario de la certificación de Java Programmer. Es curioso que aunque usamos habitualmente JDBC nuestro conocimiento sobre él es en muchos casos superficial. Hoy hablaremos sobre los tipos de ResultSets que existen, para ello partiremos del siguiente bloque de código.

```
try (Connection conexion = getConnection();
Statement sentencia = conexion.createStatement();
ResultSet resultSet = sentencia.executeQuery(sql)) {
while (resultSet.next()) {
//gestionar resultset
}
} catch (SQLException e) {

}
```

JDBC ResultSet Types

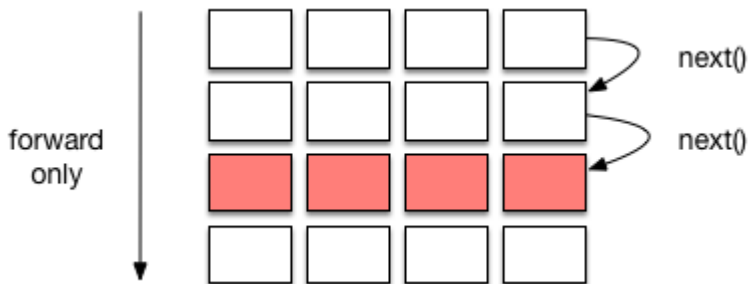
En este caso se ha usado un bloque try with resources para gestionar los recursos . Lo más importante en este código a nivel de ResultSets es la creación de un Statement :

```
Statement sentencia = conexion.createStatement();
```

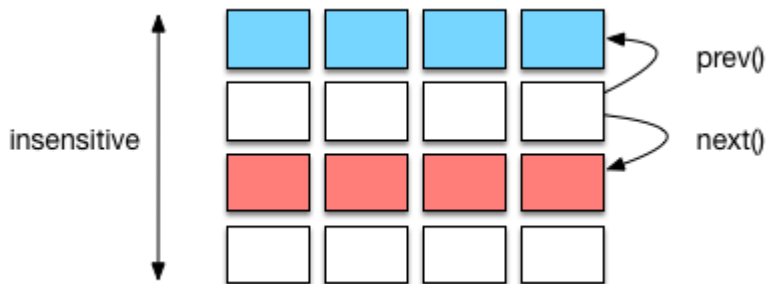
Este código es idéntico al siguiente en cuanto a funcionalidad :

```
Statement sentencia =
conexion.createStatement(ResultSet.TYPE_FORWARD_ONLY,
ResultSet.CONCUR_READ_ONLY);
```

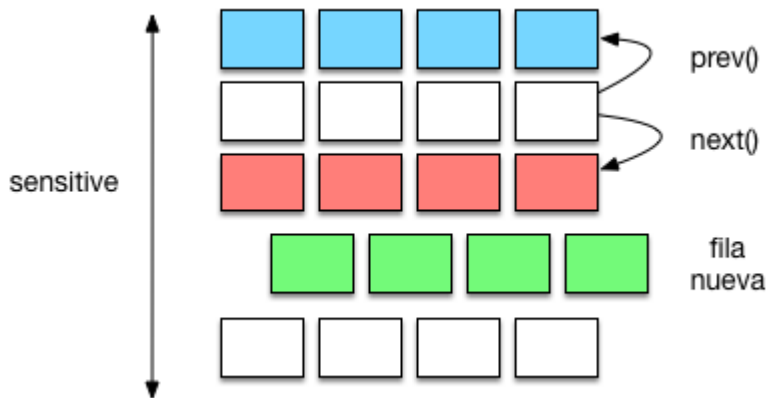
En JDBC por defecto un ResultSet es TYPE_FORWARD_ONLY y CONCUR_READ_ONLY. ¿Qué quiere decir esto? . Quiere decir que el ResultSet solo se puede mover hacia delante y solo puede leer los datos. No tiene ninguna capacidad para actualizarlos.



Existen otros dos tipos de ResultSet: TYPE_INSENSITIVE y TYPE_SENSITIVE, el primero de ellos es capaz de mover el ResultSet hacia delante y hacia atrás.



Por último **el ResultSet SENSITIVE** añade la capacidad de ver los nuevos registros que se hayan insertado en la base de datos **después de ejecutar la consulta**.



Cada uno de los tipos de ResultSet tiene su hueco dependiendo de las operaciones que uno quiera realizar. Hay que tener en cuenta que los diferentes drivers no estan obligados a soportar todos los tipos de ResultSet. Podemos asignar un ResultSet de tipo SENSITIVE y acabar con un Forward_Only ,tengamoslo en mente.

Otros artículos relacionados: [JDBC try with resources](#) , [JDBC Batch](#) , [Introducción a JPA](#)