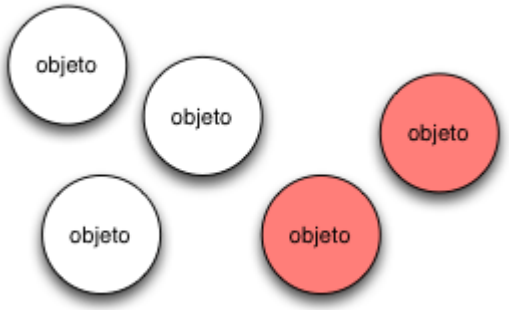
Todos hemos trabajado con JDBC en algún momento. Puede ser que hoy en día estemos usando JPA u otra cosa pero a veces toca volver, todo depende de los diferentes proyectos en los que estemos trabajando. Java 7 y Java 8 han incorporado muchas novedades pero algunas pasan un poco más desapercibidas. “Java try with resources” ha sido una de ellas que venía incorporada en Java 7 y permite una gestión del cierre de recursos mucho más sencilla . Normalmente en cualquier programa Java tenemos una serie de objetos que se construyen.



Mientras que la mayoría de ellos se pueden simplemente destruir después de haberlos usado,hay algunos que necesitan liberar una serie de recursos antes de ser destruidos .Ejemplo de este caso son Readers o Connections esto hace que el código sea complicado de gestionar. El siguiente ejemplo muestra la casuística con JDBC.

```
Connection connection = null;
PreparedStatement pStatement = null;
ResultSet resultSet = null;
try {
connection = getConnection();
pStatement = connection.prepareStatement(sql);
resultSet = pStatement.executeQuery();
```

```
while (resultSet.next()) {
// gestionar ResultSet
}
} catch (SQLException e) {
throw newException;
} finally {
if (resultSet != null) {
try {
resultSet.close();
} catch (Exception e) {
}
}
if (statement != null) {
try {
statement.close();
} catch (Exception e) {
}
}
if (connection != null) {
try {
connection.close();
} catch (Exception e) {
}
}
}
```

El código es realmente duro . Gracias a Java 7 y a la estructura de try-catch-with resources podemos simplificarlo a lo siguiente:

```
try (Connection connection = getConnection();
Prepared pStatement = connection.prepareStatement(sql);
ResultSet resultSet = pStatement.executeQuery()) {
while (resultSet.next()) {
// gestionar resultset
}
} catch (SQLException e) {
throw newException;
}
```

Esta modificación se puede utilizar con cualquier clase que implemente el interface AutoCloseable.

```
public interface AutoCloseable{

public void close();
}
```

Las ventajas son evidentes y el código es mucho mas sencillo de gestionar.

Otros artículos relacionados: Introducción a JPA ,Introducción a JTA, SQL Inyection