

CURSO DE JAVA CON JDBC

EJERCICIO

CURSORES CON CALLABLE STATEMENT DE JDBC



Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Vamos a crear un Cursor de Oracle para poner en práctica el tema de Callable Statement con JDBC. Al finalizar veremos:

The screenshot shows the NetBeans IDE environment. The Project Explorer on the left displays the project structure for 'CallableStatementCursos'. The Source Editor in the center shows the code for 'TestCursos.java'. The Output window at the bottom displays the execution results of the program.

```
package cs;

import datos.Conexion;
import java.sql.*;
import oracle.jdbc.*;

public class TestCursos {

    public static void main(String[] args) {
        //Utilizamos una clase de oracle para poder procesar el
        //cursor que regresa la funcion de Oracle
        OracleCallableStatement oraCallStmt = null;
        OracleResultSet deptResultSet = null;
        try {
            Connection con = Conexion.getConnection();
            //Tiene dos parametros que posteriormente se definiran
            oraCallStmt = (OracleCallableStatement) con.prepareCall("{? = call ref_cursor_?}");
            //Indicamos el tipo de regreso, el cual es un cursor
            oraCallStmt.registerOutParameter(1, OracleTypes.CURSOR); //Parametro 1
            oraCallStmt.setInt(2, 200); //establecemos departamento_id, parametro 2
            oraCallStmt.executeQuery();
            //Recuperamos el resultSet y lo convertimos a un tipo Oracle
            deptResultSet = (OracleResultSet) oraCallStmt.getCursor(1);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output window (CallableStatementCursos (run)):

```
run:
Id_departamento: 210, Nombre_departamento: IT Support, Ubicacion_id: 1700
Id_departamento: 220, Nombre_departamento: NOC, Ubicacion_id: 1700
Id_departamento: 230, Nombre_departamento: IT Helpdesk, Ubicacion_id: 1700
Id_departamento: 240, Nombre_departamento: Government Sales, Ubicacion_id: 1700
Id_departamento: 250, Nombre_departamento: Retail Sales, Ubicacion_id: 1700
Id_departamento: 260, Nombre_departamento: Recruiting, Ubicacion_id: 1700
Id_departamento: 270, Nombre_departamento: Payroll, Ubicacion_id: 1700
BUILD SUCCESSFUL (total time: 0 seconds)
```

A red callout bubble points to the SQL query in the code, containing the text: `departament_id > 200`.

PASO 1. COPIAMOS EL CÓDIGO DEL CURSOR DE ORACLE

Archivo paqueteCursorPackage.sql:

```
CREATE OR REPLACE PACKAGE      "REF_CURSOR_PACKAGE"  
AS  
    TYPE t_ref_cursor IS REF CURSOR;  
    FUNCTION get_dept_ref_cursor(p_dept_id INTEGER) RETURN t_ref_cursor;  
END;  
/
```

```
CREATE OR REPLACE PACKAGE BODY ref_cursor_package  
AS
```

```
    FUNCTION get_dept_ref_cursor (p_dept_id INTEGER)  
    RETURN t_ref_cursor IS
```

```
        dept_ref_cursor t_ref_cursor;
```

```
BEGIN
```

```
    OPEN dept_ref_cursor FOR
```

```
        SELECT department_id, department_name, location_id
```

```
        FROM   departments
```

```
        WHERE  department_id > p_dept_id
```

```
        ORDER BY department_id;
```

```
    RETURN dept_ref_cursor;
```

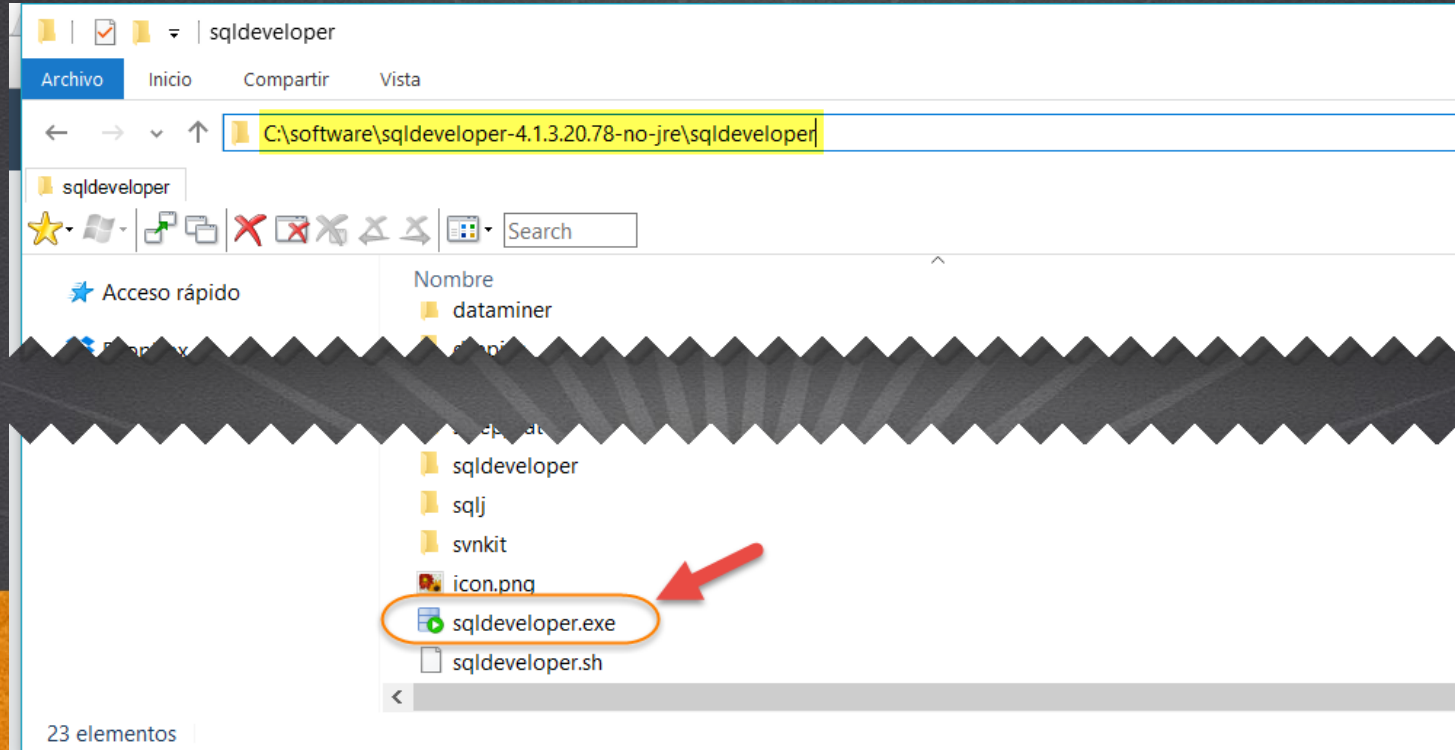
```
END get_dept_ref_cursor;
```

```
END ref_cursor_package;
```

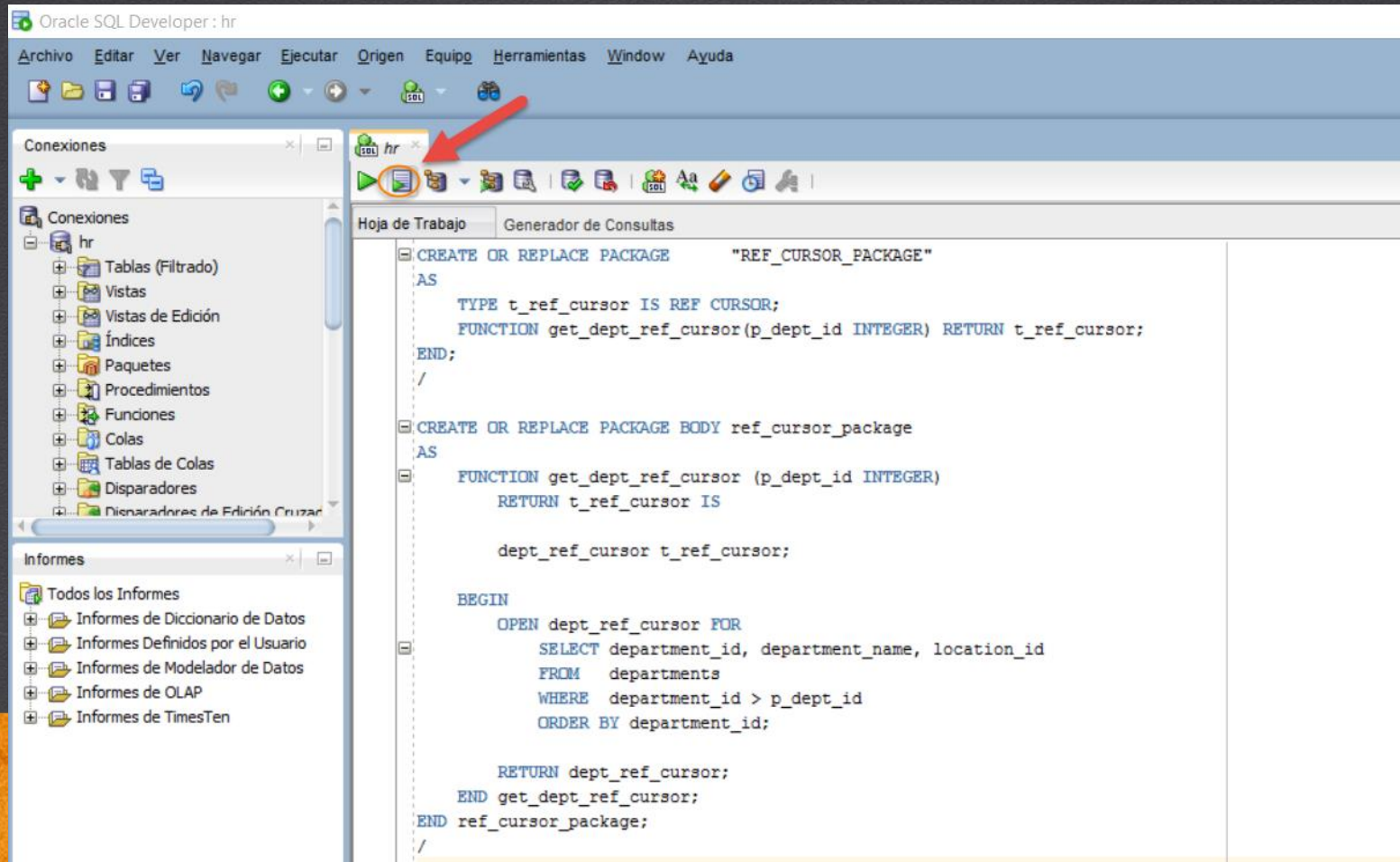
```
/
```


PASO 2. ABRIMOS SQL DEVELOPER

Abrimos el programa de SQL Developer:



PASO 3. PEGAMOS EL PROCEDIMIENTO Y LO EJECUTAMOS



PASO 4. RESULTADO DE LA EJECUCIÓN DEL PROCEDIMIENTO

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Conexiones' (Connections) pane shows a tree structure for the 'hr' schema, with 'Paquetes' (Packages) expanded to show 'REF_CURSOR_PACKAGE' and its body. Below this, the 'Informes' (Reports) pane is visible. The main workspace is divided into two panes: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Generator). The 'Generador de Consultas' pane contains the following PL/SQL code:

```
BEGIN
  OPEN dept_ref_cursor FOR
    SELECT department_id, department_name, location_id
    FROM departments
    WHERE department_id > p_dept_id
    ORDER BY department_id;

  RETURN dept_ref_cursor;
END get_dept_ref_cursor;
END ref_cursor_package;
/
```

Below the code, the 'Salida de Script' (Script Output) pane shows the execution results:

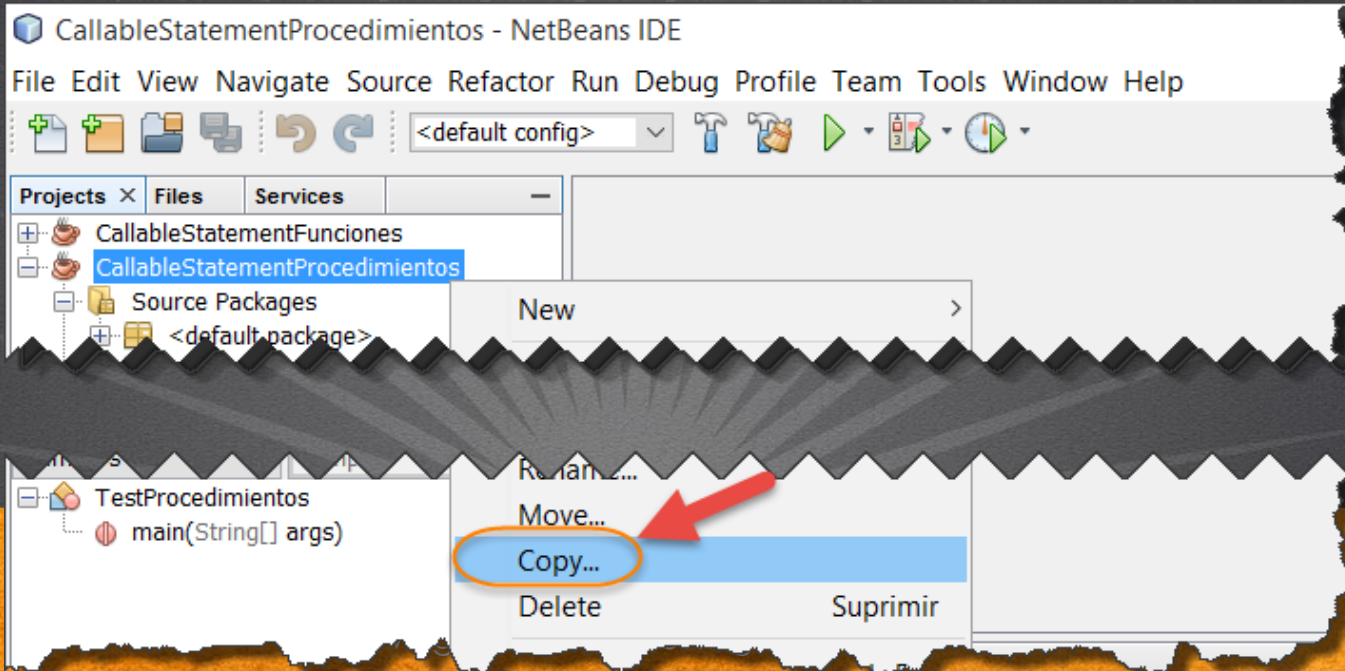
```
Tarea terminada en 0.328 segundos

Package REF_CURSOR_PACKAGE compilado

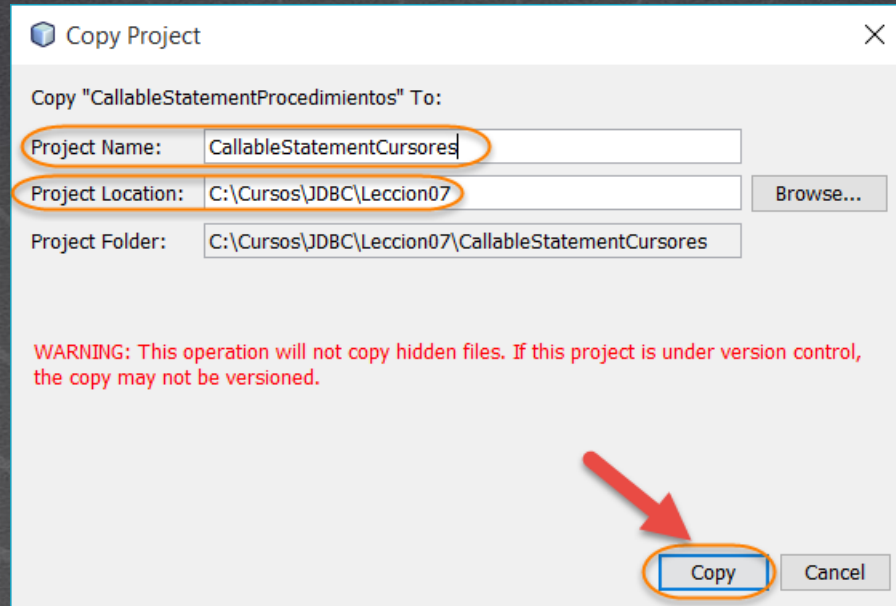
Package body REF_CURSOR_PACKAGE compilado
```


PASO 5. COPIAMOS UN PROYECTO JAVA

Copiamos el proyecto CallableStatementProcedimientos, si no es posible, solo copiamos las clases de Conexión, el archivo de propiedades jdbc y agregamos el driver de Oracle al classpath como lo hicimos en el proyecto anterior:



PASO 5. COPIAMOS UN PROYECTO JAVA



CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

PASO 6. CREAMOS UNA CLASE JAVA

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo TestCursores.java:

```
package cs;

import datos.Conexion;
import java.sql.*;
import oracle.jdbc.*;

public class TestCursores {

    public static void main(String[] args) {
        //Utilizamos una clase de oracle para poder procesar el
        //cursor que regresa la funcion de Oracle
        OracleCallableStatement oraCallStmt = null;
        OracleResultSet deptResultSet = null;
        try {
            Connection con = Conexion.getConnection();
            //Tiene dos parametros que posteriormente se definiran
            oraCallStmt = (OracleCallableStatement) con.prepareCall("{? = call
ref_cursor_package.get_dept_ref_cursor(?)}");
            //Indicamos el tipo de regreso, el cual es un cursor
            oraCallStmt.registerOutParameter(1,
OracleTypes.CURSOR); //Parametro 1
            oraCallStmt.setInt(2, 200); //establecemos departamento_id,
parametro 2
            oraCallStmt.execute();
```

```
//Recuperamos el resultSet y lo convertimos a un tipo Oracle
            deptResultSet = (OracleResultSet)
            oraCallStmt.getCursor(1);
            while (deptResultSet.next()) {
                System.out.print(" Id_departamento: " +
            deptResultSet.getInt(1));
                System.out.print(", Nombre_departamento: " +
            deptResultSet.getString(2));
                System.out.print(", Ubicación_id: " +
            deptResultSet.getString(3));
                System.out.println();
            }
            oraCallStmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

PASO 8. EJECUTAR EL CÓDIGO

CallableStatementCursores - NetBeans IDE

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services TestCursores.java

CallableStatementCursores

- Source Packages
 - <default package>
 - cs
 - TestCursores.java
 - TestFunciones.java
 - TestProcedimientos.java
 - datos
 - Conexion.java
- Libraries
 - ojdbc6.jar
 - JDK 1.8 (Default)
- CallableStatementFunciones
- CallableStatementProcedimientos

main - Navigator

Members

TestCursores

- main(String[] args)

```
1 package cs;
2
3 import datos.Conexion;
4 import java.sql.*;
5 import oracle.jdbc.*;
6
7 public class TestCursores {
8
9     public static void main(String[] args) {
10         //Utilizamos una clase de oracle para poder procesar el
11         //cursor que regresa la funcion de Oracle
12         OracleCallableStatement oraCallStmt = null;
13         OracleResultSet deptResultSet = null;
14         try {
15             Connection con = Conexion.getConnection();
16             //Tiene dos parametros que posteriormente se definiran
17             oraCallStmt = (OracleCallableStatement) con.prepareCall("{? = call ref_cursor_?}");
18             //Indicamos el tipo de regreso, el cual es un cursor
19             oraCallStmt.registerOutParameter(1, OracleTypes.CURSOR); //Parametro 1
20             oraCallStmt.setInt(2, 200); //establecemos departamento_id, parametro 2
21             oraCallStmt.execute();
22             //Recuperamos el resultSet y lo convertimos a un tipo Oracle
23             deptResultSet = (OracleResultSet) oraCallStmt.getCursor(1);
24             //Iteramos sobre el resultSet para mostrar los datos
25             while (deptResultSet.next()) {
26                 Id_departamento = deptResultSet.getInt(1);
27                 Nombre_departamento = deptResultSet.getString(2);
28                 Ubicacion_id = deptResultSet.getInt(3);
29                 System.out.println("Id_departamento: " + Id_departamento + ", Nombre_departamento: " + Nombre_departamento + ", Ubicacion_id: " + Ubicacion_id);
30             }
31         } catch (SQLException e) {
32             e.printStackTrace();
33         }
34     }
35 }
```

cs.TestCursores

main

Output

run:

```
Id_departamento: 210, Nombre_departamento: IT Support, Ubicacion_id: 1700
Id_departamento: 220, Nombre_departamento: NOC, Ubicacion_id: 1700
Id_departamento: 230, Nombre_departamento: IT Helpdesk, Ubicacion_id: 1700
Id_departamento: 240, Nombre_departamento: Government Sales, Ubicacion_id: 1700
Id_departamento: 250, Nombre_departamento: Retail Sales, Ubicacion_id: 1700
Id_departamento: 260, Nombre_departamento: Recruiting, Ubicacion_id: 1700
Id_departamento: 270, Nombre_departamento: Payroll, Ubicacion_id: 1700
BUILD SUCCESSFUL (total time: 0 seconds)
```

deparment_id > 200

PASO 9. VERIFICAMOS RESULTADO

Oracle SQL Developer : hr~1

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Window Ayuda

Conexiones

hr

Tablas (Filtrado)

- COUNTRIES
- DEPARTMENTS
- EMPLOYEES
- JOB_HISTORY
- JOBS
- LOCATIONS
- REGIONS

Vistas

Vistas de Edición

Índices

Registros

Informes

- Todos los Informes
- Informes de Diccionario de Datos
- Informes Definidos por el Usuario
- Informes de Modelador de Datos
- Informes de OLAP
- Informes de TimesTen

hr

DEPARTMENTS

hr~1

Hoja de Trabajo

Generador de Consultas

```
SELECT department_id, department_name, location_id
FROM departments
WHERE department_id > 200
ORDER BY department_id;
```

Resultado de la Consulta

Todas las Filas Recuperadas: 7 en 0.004 segundos

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
1	210	IT Support	1700
2	220	NOC	1700
3	230	IT Helpdesk	1700
4	240	Government Sales	1700
5	250	Retail Sales	1700
6	260	Recruiting	1700
7	270	Payroll	1700

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio pusimos en práctica el concepto de Callable Statement, y en particular cómo ejecutar una función que regresa un cursor.
- Un cursor básicamente es un tipo de Oracle, el cual es el resultado de ejecutar una consulta pero directamente en la base de datos, así que en lugar de ejecutar la lógica de negocio desde Java, se puede mandar a ejecutar desde la base de datos. Incluso habrá ocasiones en que únicamente podremos ejecutar este tipo de funciones desde Java, y será la única forma de obtener la información de la base de datos, por ello es importante saber cómo procesar e invocar funciones, procedimientos almacenados y los tipos cursor de Oracle como hemos visto.
- Otras bases de datos tienen otras formas de ejecutar estos tipos, sin embargo, es muy similar en otras bases de datos.

CURSO ONLINE

JAVA CON JDBC

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx