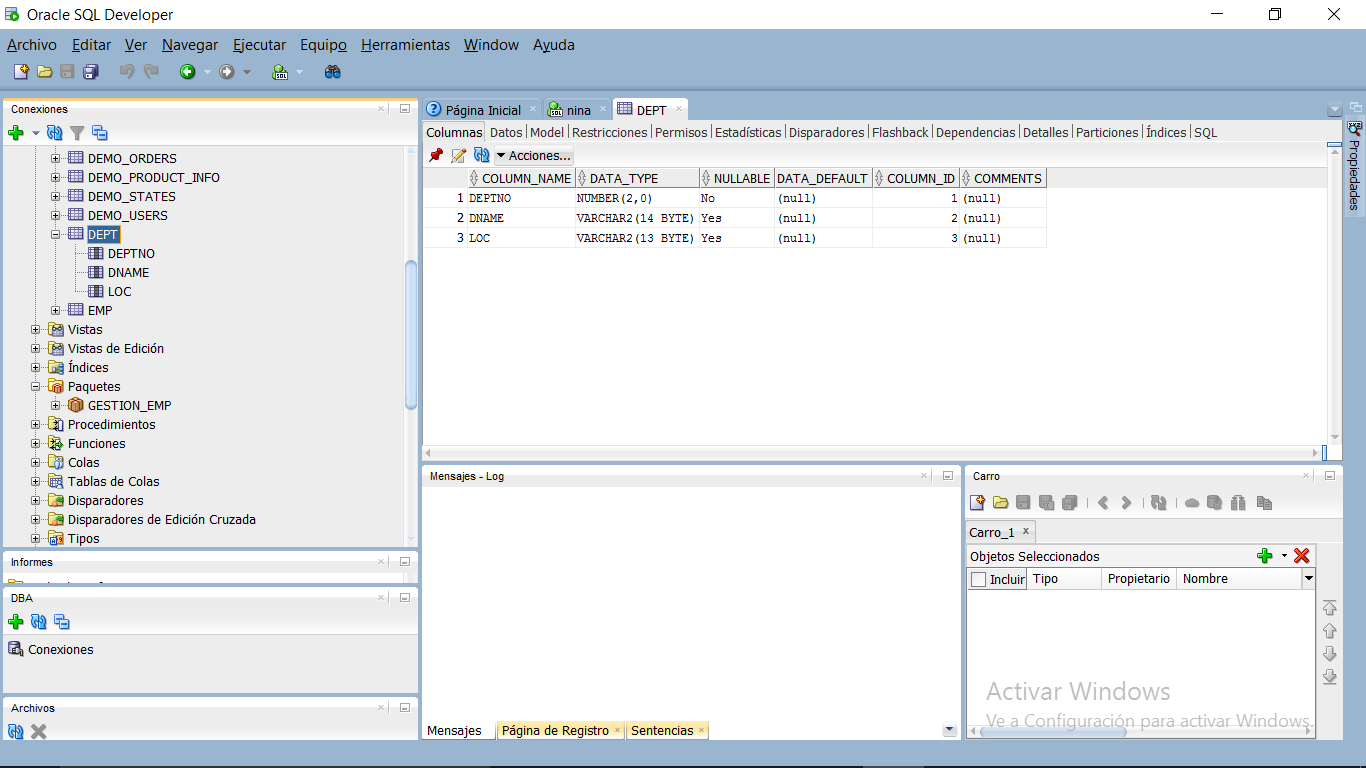
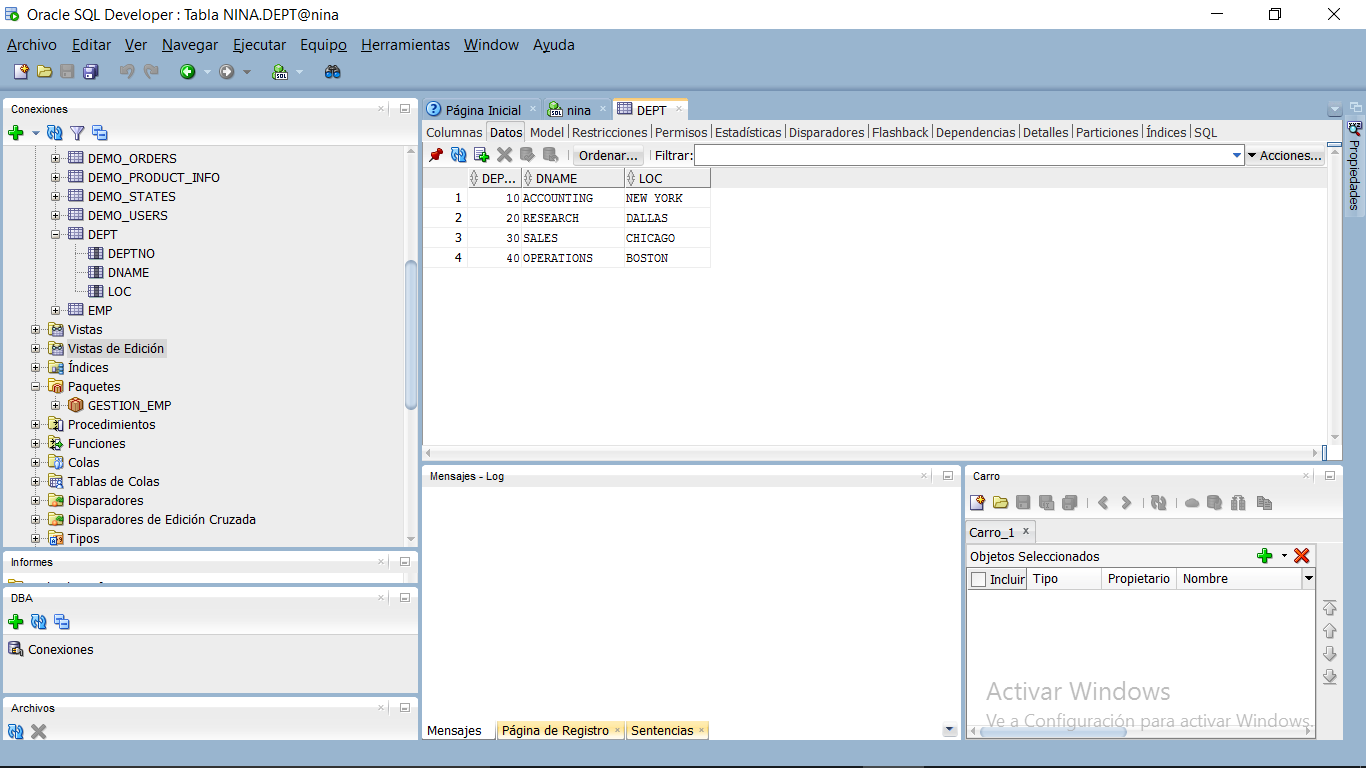
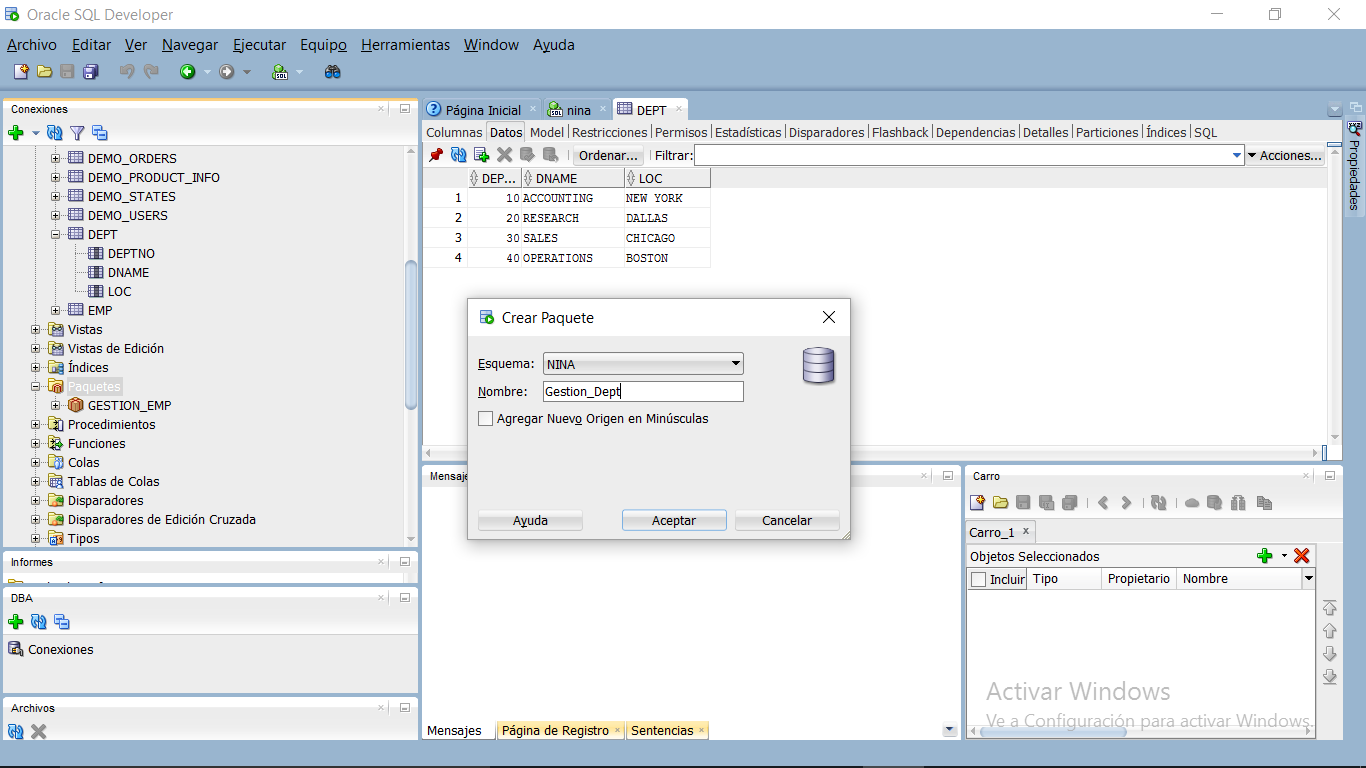
1. Dada la tabla DEPT, incluida por defecto por ORACLE en el esquema de usuario.

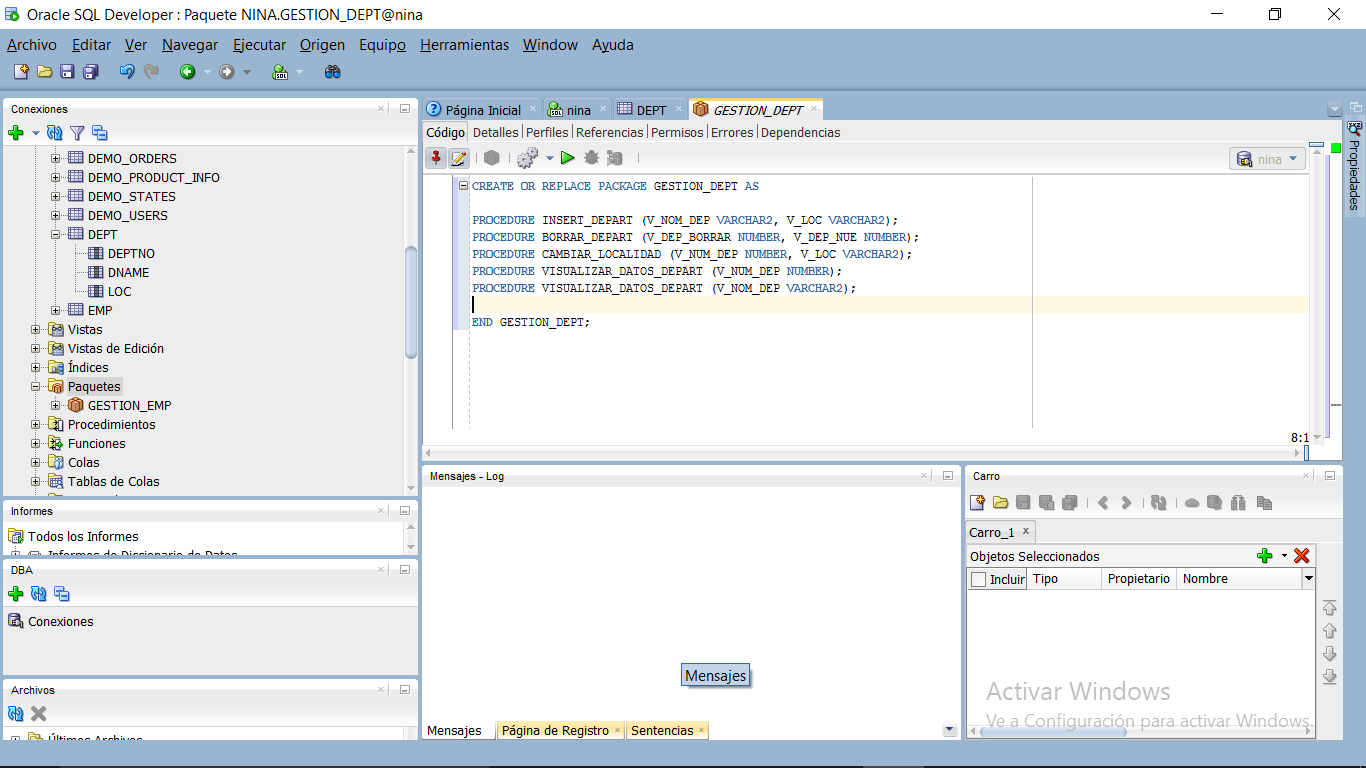




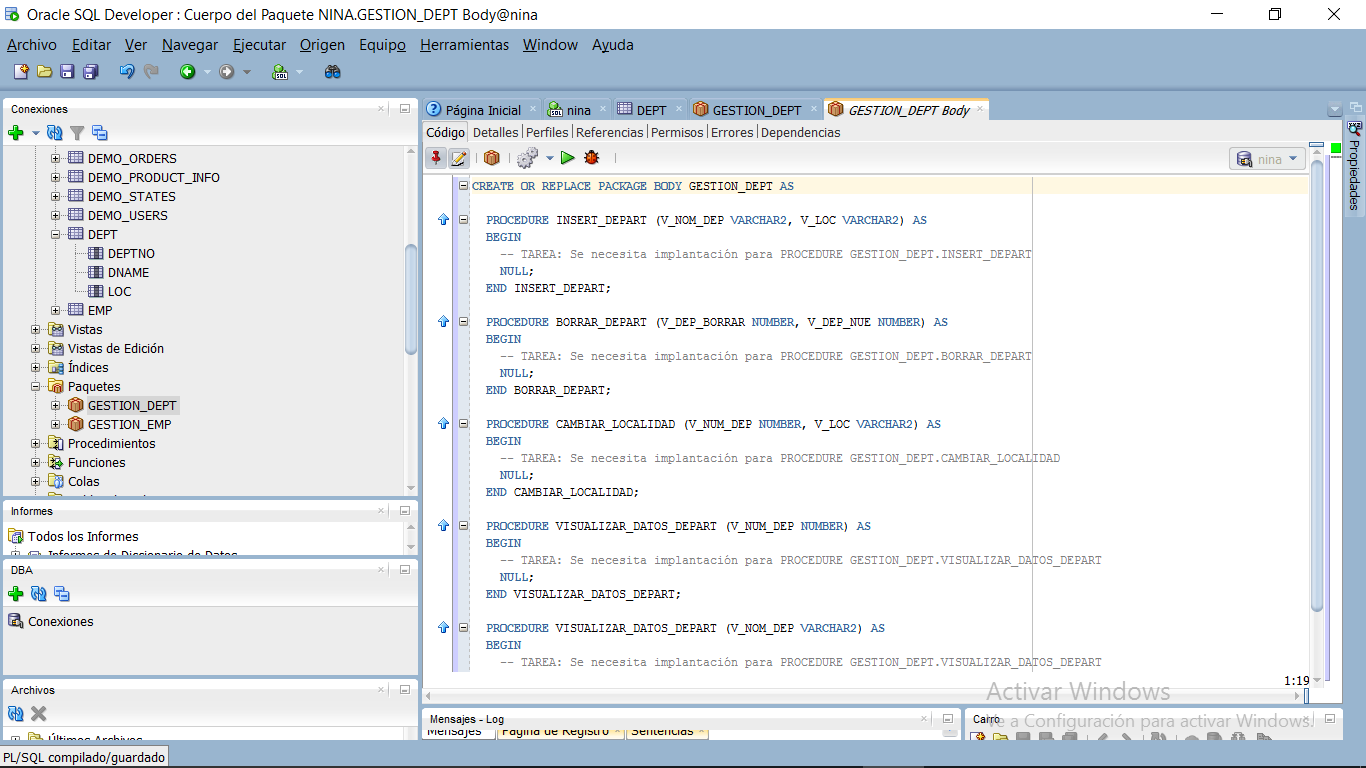
1. Escribe un paquete PLSQL para gestionar la tabla DEPT de nombre gestiónDepartamento



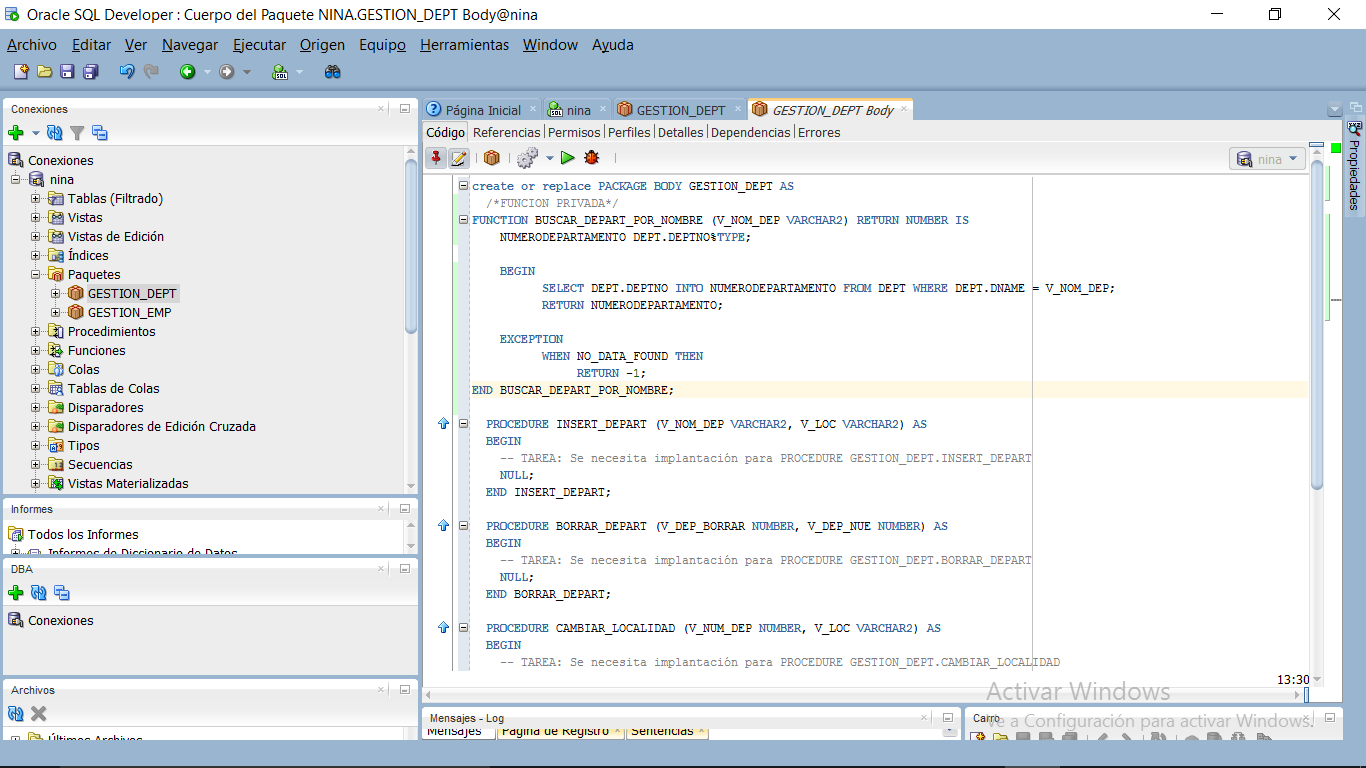
1. El paquete debe contener los siguientes subprogramas o métodos que definen la cabecera, que debe ser compilado para aparezca dentro de la carpeta package del navegador de usuario.



1. Pulsa botón derecho del ratón sobre el paquete creado y elige la opción crear cuerpo del paquete.



1. Además de los métodos anteriores, vamos a incluir una función privada Buscar\_depar\_por\_nombre



1. Codifica los procedimientos y funciones del paquete siguiendo los siguientes enunciados.

- ***INSERT\_DEPART***: permite insertar un departamento nuevo. El procedimiento recibe el nombre y la localidad del nuevo departamento. Creará el nuevo departamento comprobando que el nombre no se duplique y le asignará como número de departamento la decena siguiente al último número de departamento utilizado.

- ***BORRAR\_DEPART***: permite borrar un departamento. El procedimiento recibirá dos números de departamento de los cuales el primero corresponde al departamento que queremos borrar y el segundo al departamento al que pasarán los empleados del departamento que se va eliminar. El procedimiento se encargará de realizar los cambios oportunos en los números de departamento de los empleados correspondientes.

- ***CAMBIAR\_LOCALIDAD***: modifica la localidad del departamento. El procedimiento recibirá el número del departamento a modificar y la nueva localidad, y realizará el cambio solicitado.

- ***VISUALIZAR\_DATOS\_DEPART***: visualizará los datos de un departamento cuyo número se pasará en la llamada. Además de los datos relativos al departamento, se visualizará el número de empleados que pertenecen actualmente al departamento.

- ***VISUALIZAR\_DATOS\_DEPART***: versión sobrecargada del procedimiento anterior que, en lugar del número del departamento, recibirá el nombre del departamento. Realizará una llamada a la función buscar\_depart\_por\_nombre que se indica en el apartado siguiente.

- ***BUSCAR\_DEPART\_POR\_NOMBRE***: función local al paquete. Recibe el nombre de un departamento y devuelve el número del mismo.

1. Construye en un script, un bloque PLSQL para llamar a los subprogramas del package anterior.
2. Crea un proyecto en Netbeans GestionDep para gestionar la tabla dept. El programa debe conectar con el esquema de usuario de la base de datos Oracle y llamar a los procedimientos y funciones almacenados en el package creado anteriormente.

Instrucciones para conectar con la base de datos y realizar la llamada a un procedimiento:

**//conectar con la base de datos oracle**

**Class.forName("oracle.jdbc.driver.OracleDriver");**

**Connection conexion = (Connection) DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");**

**//construir llamada al procedimiento INSERT\_DEPART del paquete //GESTIONDEPARTAMENTO y el esquema GESTIONVENTAS**

**String sql="{call GESTIONVENTAS.GESTIONDEPARTAMENTO.INSERT\_DEPART(?,?)}";**

**//preparar la llamada**

**CallableStatement llamada=conexion.prepareCall(sql);**

**//dar valor a los parámetros**

**llamada.setString(1,"INDUSTRIALES");**

**llamada.setString(2, "MADRID");**

**//ejecutar el procedimiento**

**llamada.executeUpdate();**

**//cerrar flujos**

**llamada.close();**

**conexion.close();**

**LLAMADA A UNA FUNCIÓN**

**?=call nombre\_función (?,?,… )**

**//Por ejemplo si la función BUSCAR\_DEPAR\_POR\_NOMBRE del paquete no fuera interna, la podríamos llamar**

**String sql="{? = call GESTIONVENTAS.GESTIONDEPARTAMENTO. BUSCAR\_DEPAR\_POR\_NOMBRE (?)}";**

**//preparar la llamada**

**CallableStatement llamada=conexion.prepareCall(sql);**

**//para obtener el valor devuelto por la función**

**Llamada.registerOutParameter(1, Types.NUMERIC);**

**//dar valor al parámetro de entrada**

**llamada.setString(2,"INDUSTRIALES");**

**//ejecutar la función**

**llamada.executeUpdate();**

**System.out.println(“El número de departamento es”+llamada.getInt(1));**

**//cerrar flujos**

**llamada.close();**

**conexion.close();**

**Para visualizar desde Java las líneas de datos generadas utilizando el package dbms\_output**

**//Por ejemplo para llamar al procedimiento VISUALIZAR\_DATOS\_DEPART**

**//construimos la llamada**

**String sql = "{CALL GESTIONVENTAS.GESTIONDEPARTAMENTO.VISUALIZAR\_DATOS\_DEPART(?)}";**

**//preparar la llamada**

**CallableStatement llamada = conexion.prepareCall(sql);**

**//dar valor a los parámetros**

**llamada.setString(1, "INDUSTRIALES");**

**enable\_dbms\_output(conexion, 100);**

**//ejecutar el procedimiento**

**llamada.execute();**

**print\_dbms\_output(conexion);**

**//cerrar flujos**

**llamada.close();**

**conexion.close();**

**//Método enable\_dbms\_output Activa la salida dbms\_output, declara el tamaño del buffer y almacena las líneas en este.**

**public static void enable\_dbms\_output(Connection conn, int buffer\_size)**

**{**

**try {**

**//Activa la salida dbms\_output  
      CallableStatement stmt = conn.prepareCall("{call sys.dbms\_output.enable(?) }");**

**//almacena las líneas de salida en el buffer  
            stmt.setInt(1, buffer\_size);  
            stmt.execute();  
             
        } catch (Exception e) {  
            System.out.println("Error al activar dbms\_output! " + e.toString());}}**

**//Método print\_dbms\_output escribe la información almacenada en el buffer**

**public static void print\_dbms\_output(Connection conn)**

**{  
 try {**

**//obtiene las líneas almacenadas en el buffer  
      CallableStatement stmt = conn.prepareCall("{call  
sys.dbms\_output.get\_line(?,?)}");**

**//el primer parámetro de salida (out) representa una línea.**

**//el segundo parámetro de salida representa el estado**

**stmt.registerOutParameter(1, Types.VARCHAR);  
      stmt.registerOutParameter(2, Types.NUMERIC);**

**String line;  
      stmt.execute();**

**while((line = stmt.getString(1)) != null) {**

**System.out.println(" " + line);  
                stmt.execute();**

**}**

**} catch (Exception e) {  
            System.out.println("ERROR+ e.toString());  
        }  
    }**