



CICLO: [DAM]
MÓDULO DE [AD]

[Tarea N° Global]

Alumno:
[Jairo Martínez Garrido]

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2018 Fomento Ocupacional FOC SL todos los derechos reservados.

Contenido

1. Documentos que se adjuntan a este informe.....	2
2. MigracionXML.	2
3. MigracionBDOO.	6

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Proyecto NetBeans

2. MigracionXML.

Conexión con Oracle

The screenshot shows the 'Nueva / Seleccionar Conexión a Base de Datos' dialog box. The 'Nombre de Cone...' tab is selected. The 'Name' field contains 'schema-hr'. The 'Tipo de Base de Datos' is set to 'Oracle'. The 'Información de usuario' tab is active, showing 'Tipo de autenticación' set to 'Por defecto', 'Usuario' as 'HR', and 'Contraseña' as '*****'. The 'Rol' is 'valor por defecto' and 'Guardar Contraseña' is checked. The 'Tipo de Conexión' is 'Básico'. The 'Detalles' tab is active, showing 'Nombre del Host' as 'localhost', 'Puerto' as '1521', and 'SID' as 'xe'. The 'Nombre del Servicio' is empty. The 'Estado:' field is empty. At the bottom are buttons for 'Ayuda', 'Guardar', 'Borrar', 'Probar', 'Conectar', and 'Cancelar'.

Actions:
 Removing collection: /db/system ...
 Removing collection: /db/twitter ...

Browsing Collection: /db

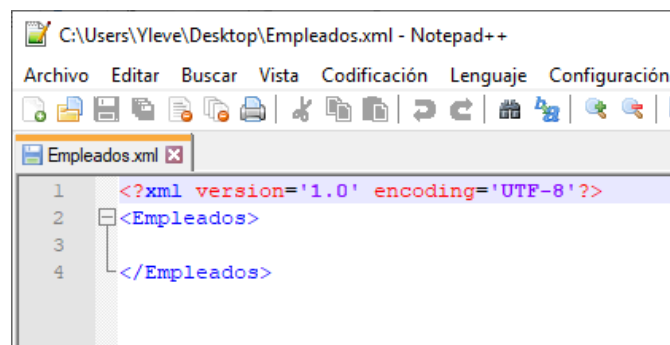
Name	Permissions	Owner	Group	Created	Modified	Size (KB)	Revision
Up							
<input type="checkbox"/> Empleados	rwur-ur-u	admin	dba	May 26 2019 22:10:55			

New collection:

No se eligió archivo

Creamos una colección **Empleados**

Y un fichero **empleados.xml** que subiremos a la base de datos eXist



Una vez realizados estos pasos de preparación para la tarea pasamos al código.

```
//Conexion JDBC con ORACLE
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection conexion = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","HR","oracle");
```

Establecemos la conexión con la base de datos Oracle y el esquema HR.

```
//Conexion XML:DB
Class driverClass = Class.forName("org.exist.xmlldb.DatabaseImpl");
Database db =(Database) driverClass.newInstance();
DatabaseManager.registerDatabase(db);

//Cargar coleccion
String url= "xmlldb:exist://localhost:8080/exist/xmlrpc/db/empleados";
String user = "admin";
String password = "123456";
org.xmlldb.api.base.Collection colec = DatabaseManager.getCollection(url, user, password);
```

Establecemos la conexión con la base de datos **eXist** y cargamos la colección **Empleados**.

```
if(colec != null){
    //Creacion del servicio
    XPathQueryService servicio = (XPathQueryService) colec.getService("XPathQueryService","1.0");

    //Consulta todos los empleados
    Statement sentencia = conexion.createStatement();
    ResultSet resultado = sentencia.executeQuery("SELECT * FROM Employees");
```

Si la colección que hemos recuperado no está vacía, creamos el servicio que nos permitirá realizar la inserción en la base de datos eXist. También creamos la consulta a la base de datos de Oracle para recuperar toda la información de la tabla HR.

```
//Mostrar por pantalla todos los empleados
while(resultado.next()){
    //ID, FirstName, LastName, Email, PhoneNumber, HireDate, JobID, Salary, COMMISSION_PCT, Manager_ID, Department_ID
    String sentenciaInsercion ="update insert <EMPLEADO>" +
        "\n\t<EMP_ID>"+resultado.getInt(1)+"</EMP_ID>\n" +
        "\t<FIRST_NAME>"+resultado.getString(2)+"</FIRST_NAME>\n" +
        "\t<LAST_NAME>"+resultado.getString(3)+"</LAST_NAME>\n" +
        "\t<EMAIL>"+resultado.getString(4)+"</EMAIL>\n" +
        "\t<PHONE_NUM>"+resultado.getString(5)+"</PHONE_NUM>\n" +
        "\t<HIRE_DATE>"+resultado.getDate(6)+"</HIRE_DATE>\n" +
        "\t<JOB_ID>"+resultado.getString(7)+"</JOB_ID>\n" +
        "\t<SALARY>"+resultado.getInt(8)+"</SALARY>\n" +
        "\t<COMMISSION_PCT>"+resultado.getInt(9)+"</COMMISSION_PCT>\n" +
        "\t<MANAGER_ID>"+resultado.getInt(10)+"</MANAGER_ID>\n" +
        "\t<DEPT_ID>"+resultado.getInt(11)+"</DEPT_ID>\n" +
        "</EMPLEADO> into /Empleados";

    servicio.query(sentenciaInsercion);
```

Mientras la consulta que hacemos ala base de datos Oracle tenga resultados que recorrer, los recuperamos e introducimos en la estructura de inserción para la base de datos eXist generando una estructura XML y lanzando la inserción finalmente, así hasta que no haya más empleados que recoger.

```
//LECTURA DEL FICHERO XML
File f = new File("C://users//Yleve//Desktop//XML//Empleados.xml");

FileReader fr = new FileReader(f);
BufferedReader br = new BufferedReader(fr);
String line = br.readLine();
System.out.println("LECTURA DE FICHERO XML EMPLEADOS.XML");
//Lector del fichero
while(line != null){
    System.out.println(line);
    line = br.readLine();
}
br.close();
```

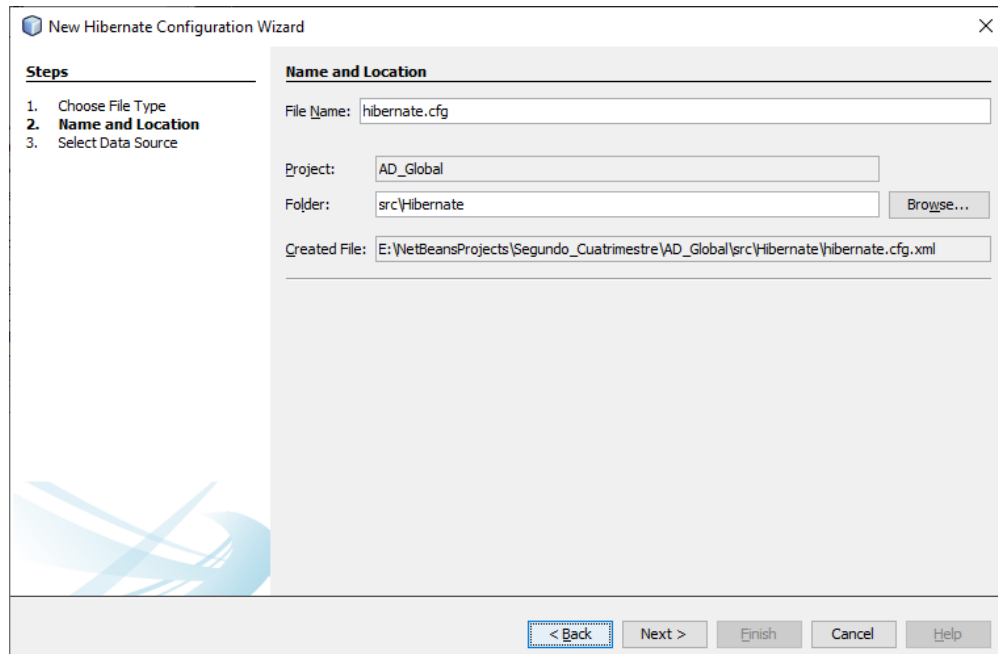
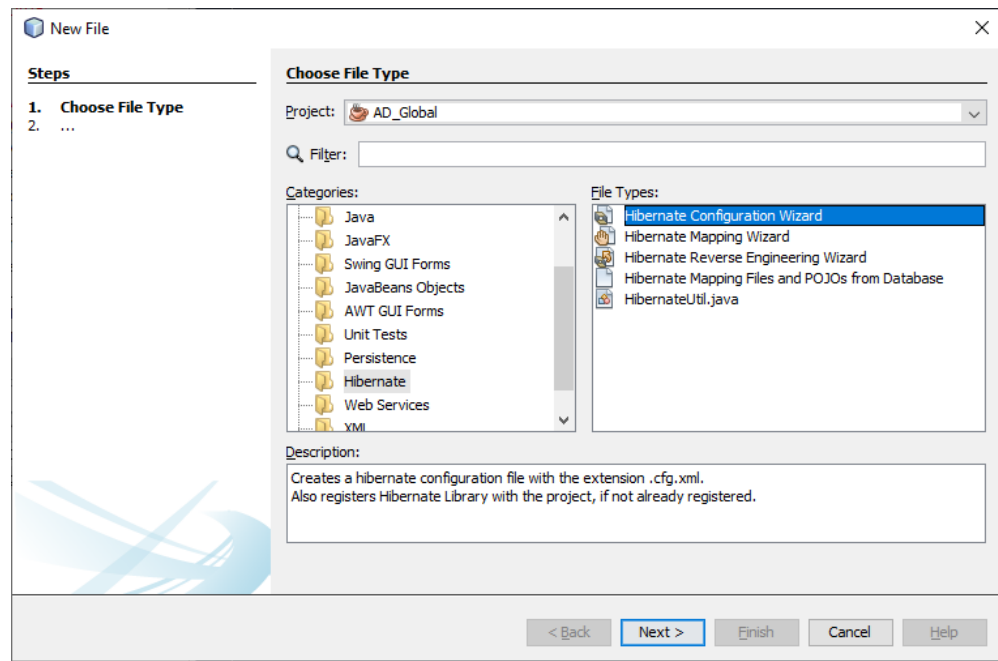
A continuación, realizamos la consulta del fichero que hemos generado mediante la lectura de este e imprimiendo por consola cada línea que recuperamos.

```
run:
LECTURA DE FICHERO XML EMPLEADOS.XML
<Empleados>

  <EMPLEADO>
    <EMP_ID>100</EMP_ID>
    <FIRST_NAME>Steven</FIRST_NAME>
    <LAST_NAME>King</LAST_NAME>
    <EMAIL>SKING</EMAIL>
    <PHONE_NUM>515.123.4567</PHONE_NUM>
    <HIRE_DATE>2003-06-17</HIRE_DATE>
    <JOB_ID>AD_PRES</JOB_ID>
    <SALARY>24000</SALARY>
    <COMMISSION_PCT>0</COMMISSION_PCT>
    <MANAGER_ID>0</MANAGER_ID>
    <DEPT_ID>90</DEPT_ID>
  </EMPLEADO>
  <EMPLEADO>
    <EMP_ID>101</EMP_ID>
    <FIRST_NAME>Neena</FIRST_NAME>
    <LAST_NAME>Kochhar</LAST_NAME>
    <EMAIL>NKOCHHAR</EMAIL>
    <PHONE_NUM>515.123.4568</PHONE_NUM>
    <HIRE_DATE>2005-09-21</HIRE_DATE>
    <JOB_ID>AD_VP</JOB_ID>
    <SALARY>17000</SALARY>
    <COMMISSION_PCT>0</COMMISSION_PCT>
    <MANAGER_ID>100</MANAGER_ID>
    <DEPT_ID>90</DEPT_ID>
  </EMPLEADO>
</Empleados>
```

3. MigracionBDOO.

Instalación de Hibernate



Configuramos el fichero config y su fuente de datos

New Connection Wizard

Customize Connection

Driver Name: Oracle Thin (Service ID (SID))

Host: localhost Port: 1521

Service ID (SID): XE

User Name: HR

Password: ●●●●●●

☐ Remember password

Connection Properties Test Connection

JDBC URL: jdbc:oracle:thin:@localhost:1521:XE

< Back Next > Finish Cancel Help

New Hibernate Configuration Wizard

Steps

1. Choose File Type
2. Name and Location
3. **Select Data Source**

Select Data Source

Database Connection: jdbc:oracle:thin:@localhost:1521:XE [HR on HR]

Database Dialect: org.hibernate.dialect.OracleDialect

< Back Next > Finish Cancel Help

Ingeniería inversa

New Hibernate Reverse Engineering Wizard

Steps

1. Choose File Type
2. Name and Location
3. **Database Tables**

Database Tables

Configuration File:

Available Tables:

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

Selected Tables:

☒ Include Related Tables

< Back Next > **Finish** Cancel Help

New Hibernate Reverse Engineering Wizard

Steps

1. Choose File Type
2. Name and Location
3. **Database Tables**

Database Tables

Configuration File:

Available Tables:

Selected Tables:

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

☒ Include Related Tables

< Back Next > **Finish** Cancel Help

POJOS

New Hibernate Mapping Files and POJOs from Database

Steps

1. Choose File Type
- 2. Generation of Code**

Generation of Code

Hibernate Configuration File:

Hibernate Reverse Engineering File:

General Settings:

- ☐ JDK 5 Language Features
- ☐ EJB 3 Annotations

Code Generation Settings:

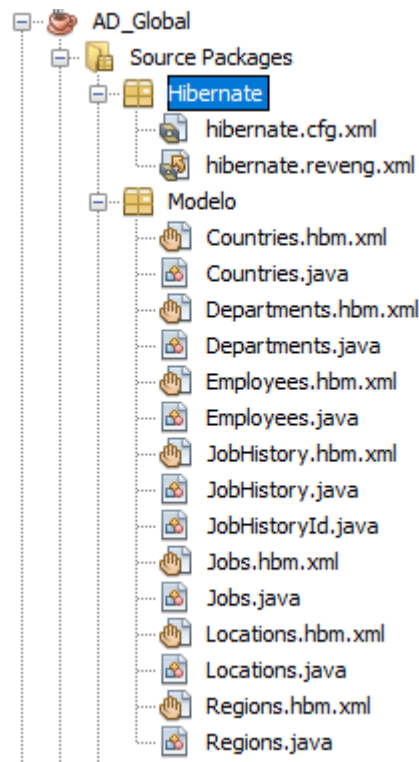
- ☒ Domain Code (.java)
- ☒ Hibernate XML Mappings (.hbm.xml)

Project:

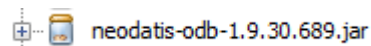
Location:

Package:

< Back Next > **Finish** Cancel Help



Importamos la librería de **Neodatis**



```
public class MigracionBDOO {
    public static void main(String[] args) {
        borrarFichero();
        //creamos la sesión consulta Hibernate
        SessionFactory sesionGeneral = SessionFactorySingleton.getSessionFactory();
        Session sesionActual = sesionGeneral.openSession();

        MigracionBDOO programa = new MigracionBDOO();
        programa.iniciar(sesionActual);
    }
}
```

Main de **MigracionBDOO** donde iniciamos la ejecución del programa mediante el método **iniciar()**. Previamente creamos la conexión con **Hibernate** y pasamos como parámetro el objeto **Session** al método iniciar.

```

/**
 * Borra el fichero de la base de datos para cuando iniciemos el programa
 */
public static void borrarFichero() {
    File f = new File("C://users//Yleve//Desktop//Employees.Neodatis");
    if(f.exists())
        f.delete();
}

```

Creamos un método **borrarFichero** para asegurarnos que al ejecutar el programa no haya ningún fichero con el mismo nombre que el de la base de datos que vamos a utilizar.

```

public static void iniciar(Session sesionActual) {

    //Crear Neodatis
    ODB odb = ODBFactory.open("C://users//Yleve//Desktop//Employees.Neodatis");

    //Consulta en Hibernate
    Query q = sesionActual.createQuery("from Employees");

    //Creamos la lista de empleados
    List<Employees> listaEmpleados = q.list();

    //Almacenamos los elementos en Neodatis
    for (Employees empleado: listaEmpleados) {
        odb.store(empleado);
    }

    org.neodatis.odb.Objects<Employees> listaEmployees = odb.getObjects (Employees.class);
    Employees emp;
    //Recorremos la lista para mostrar los empleados
    while (listaEmployees.hasNext()) {
        emp = listaEmployees.next();
        System.out.println("*****");
        System.out.println("ID: "+emp.getEmployeeId()+"\n"
            +"Nombre: "+emp.getFirstName()+"\n"
            +"Apellido: "+emp.getLastName()+"\n"
            +"Email: "+emp.getEmail()+"\n"
            +"Telefono: "+emp.getPhoneNumber()+"\n"
            +"Fecha Contratacion: "+emp.getHireDate()+"\n"
            +"Salario: "+emp.getSalary()+"\n"
            +"Comision "+emp.getCommissionPct()+"\n");
    }
    //Cerramos flujos
    odb.close();
    sesionActual.close();
}

```

Establecemos la conexión con **Neodatis** y realizamos la consulta general sobre la tabla **Employees** de con **Hibernate**. Guardamos los resultados de la consulta en **listaEmpleados** y guardamos los objetos mediante **store()** en Neodatis.

Recogemos los objetos de la consulta sobre Neodatis en **listaEmployees** de la clase **Employees**. A continuación, recorreremos en un bucle **while** la lista e imprimimos los elementos que hemos recogido.

Al final cerramos los flujos con Hibernate y Neodatis.