

## M6-UF2

### Proyecto Base de Datos Orientada a Objetos

#### Introducción

Objetivo: practicar el acceso a datos en bases de datos orientada a objetos. Averiguar cómo podemos resolver consultas simples, modificaciones, inserciones y eliminaciones en bdd orientada a objetos.

Fecha tope de entrega: 25 - 02 -18

Evaluación:

- Trabajo individual
- Requisitos
  - Programa robusto
  - Estructurado
  - Gestión de errores no obligatoria pero puntuará positivamente
- Entregar
  - Código fuente comentado. Indica al inicio de cada función/método la finalidad del mismo
  - Capturas de ejecución. Ejecutar el código con diferentes parámetros de entrada.
- Plantilla de corrección:
  - Primera parte → 2 ptos →
    - Creación de las clases. Tipos. Limitaciones. Métodos
    - Creación de la bdd
  - Segunda parte → 2 ptos →
    - Inserción de datos
    - Mostrar datos
  - Tercera parte → 2 puntos
    - Resolución de consultas
  - Cuarta parte → 2 puntos
    - Recogida de datos desde “mysql”
    - Inserción realizando las comprobaciones adecuadas.
  - Quinta parte → 2 puntos
    - Modificación salario.
    - Eliminación empleado.

#### Enunciado

Partimos de las clases Depart y Emple utilizadas a lo largo de la UF en diferentes momentos. Los atributos son:

#### **Depart:**

```
private int deptNo;  
private String dnombre;  
private String loc;
```

#### **Emple:**

```
private int empNo;  
private String apellido;
```

```
private String oficio;  
private Emple dir;  
private java.sql.Date fechaAlt;  
private float comision;  
private Depart dept;
```

### Primera parte: montar la estructura de la base de datos

Define los métodos *getter* y *setter* para cada atributo y el método *toString()* para que devuelva el nombre del departamento en la clase Depart y el nombre del empleado en la clase Emple.

Define también los constructores necesarios. Todos los empleados deben tener un director. El empleado de mayor rango (el presidente) tiene como director a él mismo.

Crea una base de datos en Neodatis para guardar datos de empleados y departamentos. LLama a la bbdd "EMPRESA.DB". Hazlo como un proyecto Eclipse, dentro del proyecto crea un paquete para las clases anteriores.

### Segunda parte: funcionalidades

El programa dará a elegir entre varias funciones:

1. Una función que inserte objetos en la base de datos. Los datos serán aportados por el usuario. Se debe comprobar que no se produzcan repeticiones de empNo o de Apellido.
2. Una función para visualizar todos los departamentos y empleados de la base de datos. Por cada empleado se debe mostrar el nombre de su director y el de su departamento.

### Tercera parte: consultas

1. Apellidos de los empleados del departamento 10
2. Número de empleados del departamento de VENTAS
3. Apellido de los empleados cuyo director es FERNÁNDEZ
4. Por cada departamento el número de empleados

NOTA: la resolución de estas consultas las podéis estructurar dentro del mismo programa en el que habéis realizado la segunda parte

### Cuarta parte: migración

- Realiza una nueva clase que permita migrar datos de la base de datos "consultas\_básicas" de mysql a Neodatis. Dicha clase dispondrá de algún método o varios que:
  - Recabará el apellido o el emp\_no del empleado que el usuario desea migrar de una base a otra.
  - Se conectará a mysql mediante JDBC y obtendrá los datos de dicho empleado.
  - Insertará el empleado en "EMPRESA.DB" realizando previamente las correcciones pertinentes (podría llamar a la funcionalidad creada en la Segunda parte)

#### Quinta parte: modificación

- Añade una funcionalidad más que permita modificar el salario de un empleado que identificamos mediante su Apellido.
- Añade una funcionalidad más que permita eliminar un empleado que identificamos mediante su Apellido.